Spring 5-15-2021

Actuation and Control Methods for Individual and Swarm Multiscale Mechanical Systems

Pouria Razzaghi
prazzaghi@smu.edu

ACTUATION AND CONTROL METHODS FOR INDIVIDUAL AND SWARM MULTISCALE MECHANICAL SYSTEMS

Approved by:

Dr. Yildirim Hurmuzlu
Mechanical Engineering
Dissertation Committee Chairperson

Dr. Edmond Richer
Mechanical Engineering

Dr. Peter Raad
Mechanical Engineering

Dr. Ali Beskok
Mechanical Engineering

Dr. Joe Zoghzoghy
Bastin Solutions

ACTUATION AND CONTROL METHODS FOR INDIVIDUAL AND SWARM MULTISCALE MECHANICAL SYSTEMS

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Mechanical Engineering

by

Pouria Razzaghi

M.S., Aerospace Engineering, Sharif University of Technology, Tehran, Iran, 2015
B.S., Aerospace Engineering, Amir Kabir University of Technology, Tehran, Iran, 2012

May 15, 2021

ACKNOWLEDGMENTS

I would like to express my deepest respect, gratitude, and appreciation to my academic advisor Dr. Yildirim Hurmuzlu, which his encouragement, guidance, and motivation lightened my way. He has been a good mentor, knowledgeable academic advisor, and a forgivable father. Thanks to you, I wish I become a better person.

I would especially like to thank Professors, Dr. Edmond Richer, Dr. Peter Raad, Dr. Ali Beskok, and Dr. Joe Zoghzoghy as the supervisory committee members, for their valuable advice and comments, which have significantly helped me to improve my work.

Also, I would like to express the greatest respect, love, appreciation, and gratitude to my mother and father, Tayyebeh Mollatayefeh and Majid Razzaghi. Thank you for dedicating your lives for me, and thank you for making me who I am today. Words can't express my gratitude for everything you have done. I hope I made you proud of me. I want to thank my brother Mahdi for his unconditional love and support. I want to thank my soul-mate, love of my life, Shide Bakhtiari for her continuous support, care, and love throughout my life. I wish I could be a good husband for you and able to give you back all your kindness in a way you deserve.

Finally, I would like to express my appreciation and love for all of my friends, who supported me throughout my journey.

Razzaghi, Pouria M.S., Aerospace Engineering, Sharif University of Technology
B.S., Aerospace Engineering, Amir Kabir University of Technology

Actuation and Control Methods for Individual and Swarm Multiscale Mechanical Systems

Advisor: Dr. Yildirim Hurmuzlu

Doctor of Philosophy conferred May 15, 2021

Dissertation completed April, 2021

Robotic mechanisms can be driven by different internally and externally applied inertial and magnetic actuations. These actuations are utilized to regulate the dynamics of robots and move them in different locomotion modes.

The first part of this dissertation is about using an external magnetic actuation to move a simple-in-design, small-scale robot for biomedical applications. The robots can be steered in different locomotion modes such as pivot walking and tumbling. The control design of this system consists of swarm algorithm under a global control input, and a vision based closed-loop controller to navigate in 2D environments.

Secondly, I propose a new Robust Nonlinear Quadratic Gaussian (RNQG) controller based on State-Dependent Riccati Equation (SDRE) scheme for continuous-time nonlinear systems. Existing controllers do not account for combined noise and disturbance acting on the system. The proposed controller is based on a Lyapunov function and a cost function includes; states, inputs, outputs, disturbance, and the noise acting on the system. We express the RNQG control law in the form of a traditional Riccati equation.

Real-time applications of a controller place a high computational burden on system implementation. This is mainly due to the nonlinear and complex form of the cost function. In order to solve this problem, the cost function is approximated by a weighted polynomial. The weights are found by using a least-squares technique and an offline neural network. The approximate cost function is incorporated into the controller by employing a method based on Bellman's principle of optimality. Finally, different examples are used to verify the utility

of the proposed control approaches.

Finally, a highly disturbed and uncertain inertially actuated hopping robot is presented. Different nonlinear control schemes are used to regulate the system. The control outcomes are compared, and a foundation to design a robust optimal controller is given.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

I would like to than

Chapter 1

Magnetically actuated robots

1.1 System description

Miniaturized untethered robots have a promising future in biomedical and engineering applications for their design simplicity, scalability, locomotive technique, and flexibility of control during different manipulative task. Over the last few decades, advances in research at the millimeter and centimeter length scale have brought various locomotive untethered devices with increased interest. While some developments in millirobots might not be directly relevant to biomedical applications, the technologies invented can be used in different engineering applications. In millirobots, their untethered components have all dimensions less than palm size and larger than 1 mm and macroscale forces such as bulk forces dominate their mechanics [1]. Therefore, design and locomotive techniques of millirobots have greater impact towards their successful control and actuation in biomedical and engineering applications.

In the previous studies [2, 3], a magnetically actuated untethered millirobot which is very simple in design, capable of producing hybrid surface motion modes (pivot walking, tapping, galloping, corner maneuvering etc.) and power efficient was proposed. It showed immense promise of the scalability of structure and flexibility of control to become functional in minimally accessible region during performing different bio/engineering task.

The field of microrobotics has been tremendously studied in recent years. The principles of the design of such small scale robots rely on an understanding of microscale physics, fabrication, and novel control strategies.

One of the main goal is to represent the motion of a magnetic microrobot under the effect of a stationary electromagnet system. We propose a microrobot that is a magnetized rigid body with two permanent magnets M_A and M_B. The magnetic moment M is assumed to

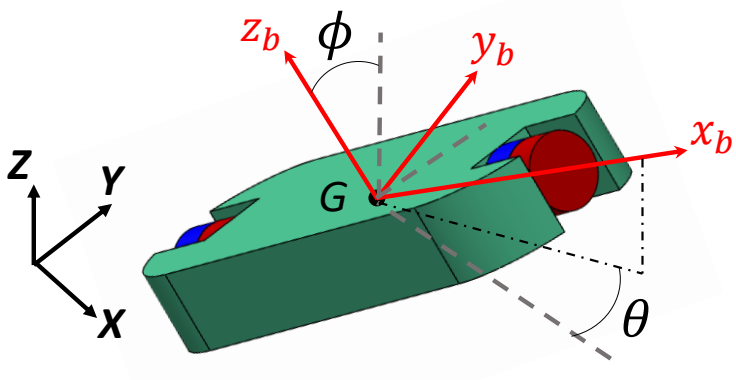


Figure 1.1: Robot configuration and the global and body reference frames.

have a constant magnitude and be rigidly connected to the frame of the body. Figure 1.1 depicts the schematic of the robot with reference frames.

The torque on the magnet can be expressed as follows:

$$T_A = M_A \times \bar{B} \tag{1.1}$$

$$T_B = M_B \times \bar{B} \tag{1.2}$$

where $\bar{B}$ is the vector of the applied magnetic field's flux density. The torque tends to align the magnetic moment with the applied field. In our case, the torque tends to align x_b axis of the body with the field.(see Fig. 1.1) In this configuration, we are able to move the center of mass of body (G) and perform rotations of the in-plane (θ) and out-of-plane (ϕ) angles. We should note that we are unable to perform rotation about the long axis of the body (x_b). The state variables of the system are $\{x, y, z, \theta, \phi\}$, where $\{x, y, z\}$ are the coordinates of the center of mass, $\{\theta, \phi\}$ are the body frame rotation angles.

The force on the magnetic moment is given as follows:

$$F_A = (M_A.\nabla)\bar{B} \tag{1.3}$$

$$F_B = (M_B.\nabla)\bar{B} \tag{1.4}$$

Each electromagnet creates a magnetic field throughout the workspace. At any given

point in the workspace P, the magnetic field due to actuating a set of given electromagnets can be expressed as follows:

$$\bar{B}(P) = \sum_{n=1}^{N} \bar{B}_n(P) = \sum_{n=1}^{N} \bar{K}_n(P) I_n \tag{1.5}$$

where subscript n represents the n^{th} actuated electromagnets, N is the number of electromagnets, I_n is the current value flowing through n^{th} electromagnet, and $\bar{K}_n(P)$ is a unit vector, thus the vector $\bar{B}(P)$ varies linearly with the current through the electromagnet. The individual field contributions are decoupled, and the fields can be individually measured and then linearly summed. The summation can be expressed in the matrix form:

$$\sum_{n=1}^{N} \bar{K}_n(P) I_n = \begin{bmatrix} \bar{K}_1 & \bar{K}_2 & \cdots & \bar{K}_n \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \mathbf{K}(P)\mathbf{I} \tag{1.6}$$

where $\mathbf{K}(P)$ is the precomputed matrix at each point in workspace of magnetic flied. To find the magnetic force, we need to find the derivative of the magnetic field in specific direction. The derivative in specific e-direction can be expressed as follows:

$$\frac{\partial \bar{B}(P)}{\partial e} = \begin{bmatrix} \frac{\partial \bar{K}_1}{\partial e} & \frac{\partial \bar{K}_2}{\partial e} & \cdots & \frac{\partial \bar{K}_n 1}{\partial e} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \mathbf{K}_e(P)\mathbf{I} \tag{1.7}$$

The magnetic torque and force on the magnetic moment M at point P is directly related to the current. The use of magnetic torque and force requires knowledge of the system's pose and magnetic moment configuration. In our case, we control the direction of the magnetic flied. However, in this case, the microrobot will align with the applied field. Thus, we need to measure the microrobot's position and orientation at point P.

The field strength, rotation direction (CW/CCW), and rotation frequency of the magnetic

field can be controlled through our magnetic control system. For instance, in xy-planar control, the resultant field can be expressed as

$$\bar{B} = \mathbf{K}(P)\mathbf{I}\begin{bmatrix}\sin\psi\,\sin\omega t & \cos\psi\,\sin\omega t & \cos\omega t\end{bmatrix}^T \tag{1.8}$$

which rotates with angular velocity (ω) around the unit vector

$$\bar{n} = \begin{bmatrix}-\cos\psi & \sin\psi & 0\end{bmatrix}^T \tag{1.9}$$

which corresponds to the desired direction, and ψ is the rotation of the plane of the rotating field. For our approximate Helmholtz coil, the magnitude of the field $|\bar{K}_i|$ of a pair of coils can be modeled as:

$$\begin{aligned}|\bar{K}_i| =& \frac{\mu_0 n\, R_{coil}}{2\left(R_{coil}^2 + d^2 - 2D\,d + D^2\right)^{3/2}} \\ &+ \frac{\mu_0 n\, R_{coil}}{2\left(R_{coil}^2 + d^2 + 2D\,d + D^2\right)^{3/2}}\end{aligned} \tag{1.10}$$

where μ_0 is the permeability constant, n is the number of turns of the coils, R_{coil} is the radius of the coils, D is the distance between a pair of coils, and d is the position between the coil. At the center of the coil, which is the position we are interested in, the position d is equal to zero. For more details on the approximate Helmholtz coil configuration, refer to Cheang *et al.* [4].

1.2 Magnetic walker robot

The magnetic walker robot is rectangular-shaped with cylindrical permanent magnets embedded in each of its two length-wise ends shown in Fig. 1.1. The idea of using the magnetic parts is when the external magnetic is applied, so alignment of the field differs from that of the robot, a magnetic torque is induced on the robot until it is realigned with the field. It is worth to note that, increasing the number of the permanent magnets along the robot axis will only amplify the induced magnetic torque. Changing the magnetic field vector causes the robot to align with the same vector direction. Based on this basic concept, various locomotion modes could be generated.

1.2.1 Equation of motion

The equations of motion are derived using Newton's method. The position of the center of mass (G) of the robot (x, y, z) and two orientation angles shown in Fig. 1.1 are taken as the states of the system. The robot experiences different forces and torques; weight (mg), surface normal force (N), a friction force (F_k) in the opposite direction of the velocity, and two external magnetic torques (τ_y, τ_z). Here F_k depends on N, the sliding friction coefficient μ, and the direction of the velocity of the contact point ($\vec{e}_V$). The microrobot is assumed to be symmetric in three direction, so the moment of inertia I_m in body fixed frame $(x_b y_b z_b)$ can be given as:

$$I_m = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{1.11}$$

In the locomotion of the system, we assume that two magnetic moments M_A and M_B are acting as pivot point. The position of one of the pivot points (A) is considered as (A_x, A_y, A_z). By using the Newton's Law of motion, the equations of motion can be written as follows:

$$\begin{aligned} m\ddot{x} &= (F_k\, \vec{e}_V).\vec{i} \\ m\ddot{y} &= (F_k\, \vec{e}_V).\vec{j} \\ m\ddot{z} &= mg - N \\ I_z\ddot{\theta} &= -\tau_z + F_k \sin\gamma\, r \sin\phi \\ I_y\ddot{\phi} &= -\tau_y + F_k \cos\gamma\, r \cos\phi - N\, r \sin\phi \end{aligned}$$

where $\{i, j, k\}$ are the unit vectors of the inertial frame (XYZ), r is the distance from G to A, γ is the angle between $\vec{e}_V$ and the x_b-direction, respectively. The kinematic constraint of

the pivot point allows the derivation of the following additional equations:

$$A_x = x - r\sin\phi\cos\theta \tag{1.12}$$

$$A_y = y - r\sin\phi\sin\theta \tag{1.13}$$

$$A_z = z - r\cos\phi \tag{1.14}$$

which the second time derivative of these equations are considered as constraint equations. As the stick-slip motion in this system, we assume that $F_k = N\,\mu$. To solve the equations of motion, we realize that we have 6 unknowns $(\ddot{x}, \ddot{y}, \ddot{z}, \ddot{\theta}, \ddot{\phi}, N)$ and five equations. To resolve this paradox, we use the pinned assumption. There are different possible types of solution that can be occurred as:

- If $N < 0$, it means that the robot does not have any contact with the ground, so $N = F_k = 0$.
- If $F_k > F_{s_{max}}$, where $F_{s_{max}} = N\,\mu_s$, it means that pinned point is slipping on the surface and $\ddot{A}_z = 0$

1.3 Basic locomotion modes

Various locomotion modes can be achieved using the proposed robot. Here, we listed few possible modes.

1.3.1 Pivot walking mode

Pivot walking is achieved by alternating the direction of the magnetic field up and down simultaneously. When the magnetic field oriented in up to down configuration, the resulting force presses A down and lifts B up. Subsequently, having a pivot formed at A, a counter-clockwise magnetic torque τ_z is applied. This causes the element to rotate forward by an angle of θ in the $x - y$ plane. In the next step, the orientation of the magnetic field is reversed, and the pivot moves to B and A is lifted. A clockwise torque τ_z is applied to rotate the member by another θ about the new pivot B. Repeating this process, locomotion along a desired path is generated as shown in Fig. 1.2(a).

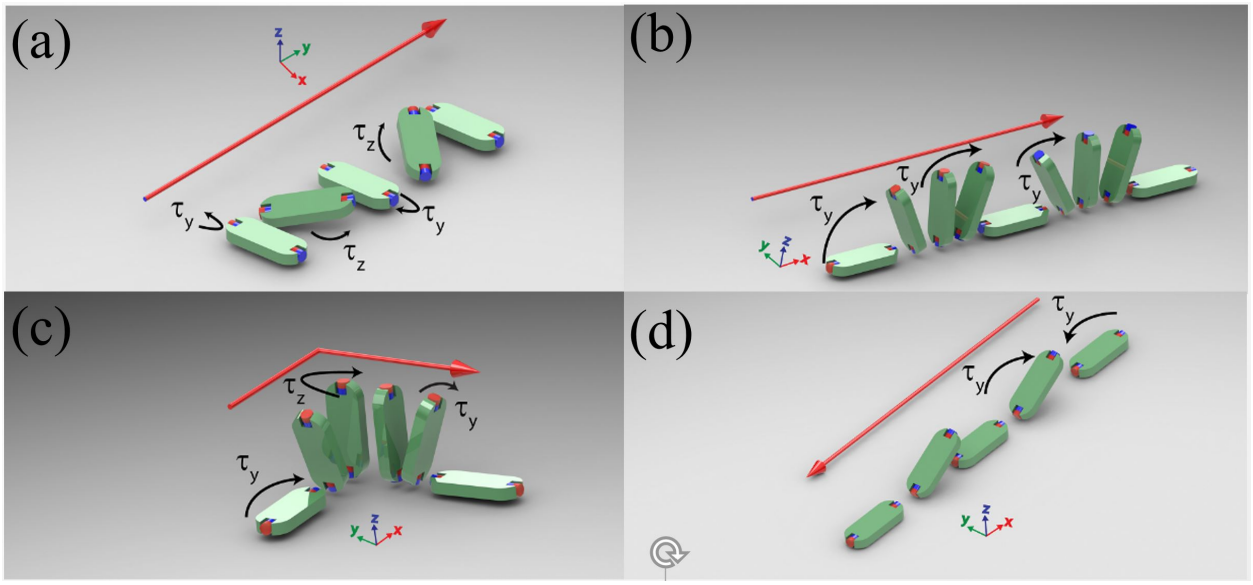


Figure 1.2: Different modes of motions. (a) Pivot walking. (b) Tumbling. (c) Corner maneuver. (d) Tapping.

1.3.2 Tumbling mode

This mode is achieved by continuously applying step torques τ_y about y axis to progress in x direction. One end will form a pivot and the whole body starts rotate about that pivot. Figure 1.2(b) shows the tumbling mode where the red arrow represents the direction of motion and the black arrows represent the direction of the applied torques.

1.3.3 Corner maneuver mode

This mode is similar to the tumbling mode, where it is achieved in similar torque profile. But when the robot reaches the upright position, a τ_z is applied to the change the orientation of the robot to progress in other direction. Figure 1.2(c) depicts the corner maneuver mode where the red arrow represents the direction of motion and the black arrows show the applied torques on each pose.

1.3.4 Tapping mode

This mode is achieved by alternating the direction of the magnetic field to form pivot at A. Then, a counter-clockwise magnetic torque τ_z is applied to orient the member on the desired direction. Finally, Applying torque in y_b repeatedly lifts and releases the trailing end,

and that will generate a forward progression in a desired direction as shown in Fig. 1.2(d).

Each mode has its own advantages and drawbacks. For the tumbling mode, its the fastest mode but it requires more space in z direction. Tapping mode has the minimum space requirement but it suffers slippage. Finally pivot walking mode is the most stable mode but on the other hand it has space limitation in x direction.

The high and significant potential of these small scaled robots encouraged us to extend our research to include incorporating position feedback using external cameras to conduct precise and more complicated experiments, and also changing the design of the robots to increase the controllability of the system. Furthermore, we will also consider swarm control algorithms and work on pattern formation of a group of these robots.

Swarm control

1.4 Problem description

Small-size robots offer access to spaces that are inaccessible to larger ones. This type of access is crucial in applications such as drug delivery, environmental detection, and collection of small samples. These robots are typically externally actuated due to their size limitation. In this case, one faces the challenge of controlling a swarm of robots using a single global input. Here, we propose a control algorithm to position individual members of a swarm in predefined positions. A single control input applies to the system and moves all robots in the same direction. Yet, we obtain another control modality by using different length robots.

First, we propose two new designs of these millirobots. In the first design, the magnets are placed at the center of body to reduce the magnetic attraction force between the millirobots. In second design, the millirobots are of identical length with two extra legs acting as the pivot points. This way we vary the pivot separation in design to take advantage of variable speed in pivot walking mode while keeping the speed constant in tumbling mode.

Then, we present a general algorithm for positional control of n millirobots with different lengths from arbitrary initial positions to final desired ones. Simulations and hardware experiments validate our results.

1.5 Introduction

Global control of a population of robots is a challenging task that requires either on-board computation [5] or a broadcast signal [6]. Swarm control of untethered small-scaled robots has recently become a popular research topic in the controls and robotics field. The size limitation of these robots makes on-board computation nearly impossible. Researchers have found ways to control groups of robots externally, such as applying a magnetic field [7]. Applying varied control inputs to individual tiny robots is also difficult. One solution is using a global control input that covers all robots. This means that a single actuation controls all robots. Moving and positioning a group of robots shows promising applications in fields such as biomedical engineering and biomechanics, particularly in drug delivery and tissue rehabilitation [8,9]. In this study, we focus on positioning a group of small-scale robots using a shared global control input.

Recently, magnetic actuation has become a standard control method for small-scale robots [10]. Yesin *et al.* [11] investigated magnetically driven millimeter-scale robots to guide them inside the human body. Miyashita and his coworkers [12, 13] developed electromagnetically actuated untethered origami robots. Although their control action resulted in walking and rolling motions, there was a high level of uncertainty in these locomotion modes. Mahoney *et al.* worked on different techniques to actuate an untethered magnetic tool such as capsule endoscopes, rolling spheres, and helical-propeller micro-swimmers [14, 15]. Son *et al.* [16] developed a localization method for an untethered magnetic robot, which was manipulated by an external electromagnetic system. Kuthan *et al.* [17] magnetically guided and actuated multiple identical robots using coplanar coils.

Applying the same control input to different robots results in the system being under-actuated. It means that we have a single input, but n degrees of freedom for n swarming robots. Other researchers have attempted to control this type of under-actuated system by adding extra constrains. These include placing obstacles in the workspace [18, 19], providing non-slip boundary contacts [20, 21], changing the physical shape of the robot [22–24], and applying an external artificial force field [25]. When there is a large number of the robots, it can be difficult to detect their individual positions, however Shahrokhi *et al.* showed that it is possible to sense global properties such as mean position and variance [6]. In this type of swarm control, a covariance ellipse was defined based on the most populated region of the workspace and the mean position is at the center [26]. Although they can place the mean position of the robots within the ellipse at the desired point, a number of robots outside the covariance ellipse can be missed or uncontrolled. Also, Dong and Sitti [27] worked on a programmable and reconfigurable system as an external static magnetic field to control the formation of micro-robots. They experimentally showed that the swarm motion of these robots can manipulate the objects, navigate through complex environments.

In our research, we control a group of robots under a global control input. The robots respond differently to the same control signal due to their different physical structures. We propose a swarm algorithm that regulates the motion of each robot individually. A different geometry (length) is utilized to add another degree of control modality to the system. The objective of swarm control is to move a group of robots from their initial to desired final

positions, in which each robot is traceable. To achieve this objective, varying the length of the robots could be useful. Each set of desired final position and number of robots requires different sets of robot lengths.

A semi elliptical-shaped millirobot is built using 3D printing. A cylindrical permanent magnet is embedded in the middle of the body. By changing the magnitude and the direction of the magnetic field vector, the millirobot can be actuated and moved in a specific direction. The motions are inspired by the inertial actuation mechanism, which was developed in our lab [28–30].

We select the pivot walking as the primary locomotion mode and the tumbling mode as the secondary one. When the global control input is applied to the millirobots, they will move parallel to each other but their velocities will be different in the pivot walking mode. The difference in their velocities would be proportional to their lengths. We exploit this feature to place an arbitrary number of millirobots, with pre-assigned lengths, at desired final locations. Also, the millirobots will move parallel and with the same velocities in the tumbling mode. This will give us an extra tool to move a swarm of millirobots. Figure 1.3 shows an example of positioning the millirobots in a pre-defined pattern in both simulation and experimental results.

To describe the swarm motion, we first present different walking trajectories performed using pivot walking mode. The kinematic equations of pivot walking are also presented. These equations are functions of the robot's length, the sweep angles of each pivot point, and the number of steps taken by the millirobots. Different walking trajectories can be obtained by varying sweep angles and number of steps. These walking trajectories are the tools of the swarm position control. Subsequently, we generalize an algorithm to swarm of n millirobots.

We propose two different designs of the millirobots. A semi-oval shape with a cylindrical permanent magnet embedded at the center of the body is the primary design (see Fig. 1.4(a)). We should note that placing the magnets at the center of the body reduces the attraction forces between the magnets that appeared in our previous millirobots [2]. The new millirobots are printed in four different lengths as 3, 5, 7, and 9 mm. In this design, the velocities of steering the millirobots are proportional to their lengths in both pivot walking

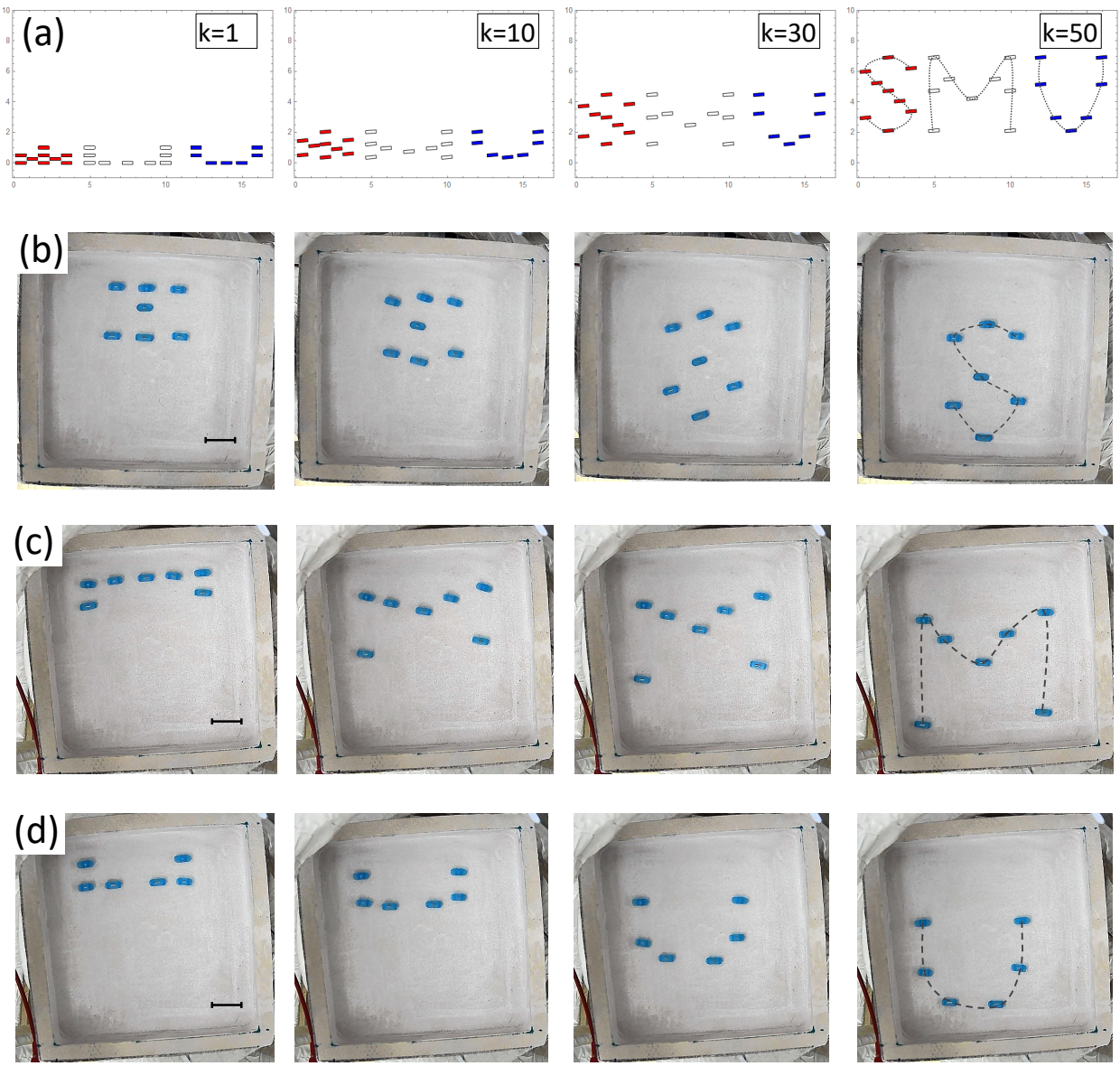


Figure 1.3: Swarm positioning motions to conduct a pre-defined pattern. **(a)** Simulation results of the positioning algorithm to generate SMU pattern. Term k denotes the number of steps **(b-d)** Experimental results of S, M, and U alphabets. The scale bars are 20 mm.

and tumbling modes of motions. In order to differentiate pivot walking and tumbling motions, we change the design by adding two legs (see Fig. 1.4(b)). In the secondary design, the lengths of the millirobots are fixed at 10 mm, but they have different pivot separations between two legs (P_s) as 3, 5, 7, and 9 mm. Thus, they can move in different velocities in the pivot walking mode, but the same velocity in the tumbling mode. This difference gives us two fundamental flexibilities in the proposed swarm pattern motion. First, we can generate a specific and desired formation employing the swarm algorithm in the pivot walking mode; then, the formation can be moved to any desired location using the tumbling mode without any changes in the final shape. The illustration of the pivot walking and tumbling motions are shown Figs. 1.4(d) and (e). Also, the directions of the magnetic field required to conduct each motion are drawn.

A low-cost high-performance magnetic actuator setup, a nested electromagnetic Helmholtz coil, is designed and constructed to actuate the presented millirobots. This system is configured based on the optimal design presented in [31]. The setup consists of three nested electromagnetic Helmholtz coils, two HP 750W power supplies, two Pololu Dual VNH5019 motor drivers, Arduino UNO, computer, camera, and pivot walker (For more details see Fig. 1.27). The three coils were powered by two HP 750W power supplies which were connected in series. The two coil drivers were used to control the power supplies and transmit the required power to the electromagnetic coils. MATLAB was used to run the main controllers and generate the required input for the coils. Also, MATLAB was used in the feedback to process the image frames from the camera video stream. Arduino board was used as a communication bridge between MATLAB and the coil drivers. Arduino UNO was also used to run the low-level control that converts the high-level commands into current command to control the coils.

The large-scale coil system produces an uniform static magnetic field, which can rotate in 3D dimension. The outer diameters of coils are 39, 30.5, and 22.5 cm in x, y, and z directions respectively. The separation distances between coil pairs are 24, 19, and 11 cm. The system has a $12\,\text{cm} \times 12\,\text{cm}$ work space at the center of the configuration (see Fig. 1.5(c)). The coils are fabricated using insulated 12 gauge circular copper wire. Figures 1.5(a) and (b) show the isometric views of the CAD drawing and the actual coil system. The maximum current

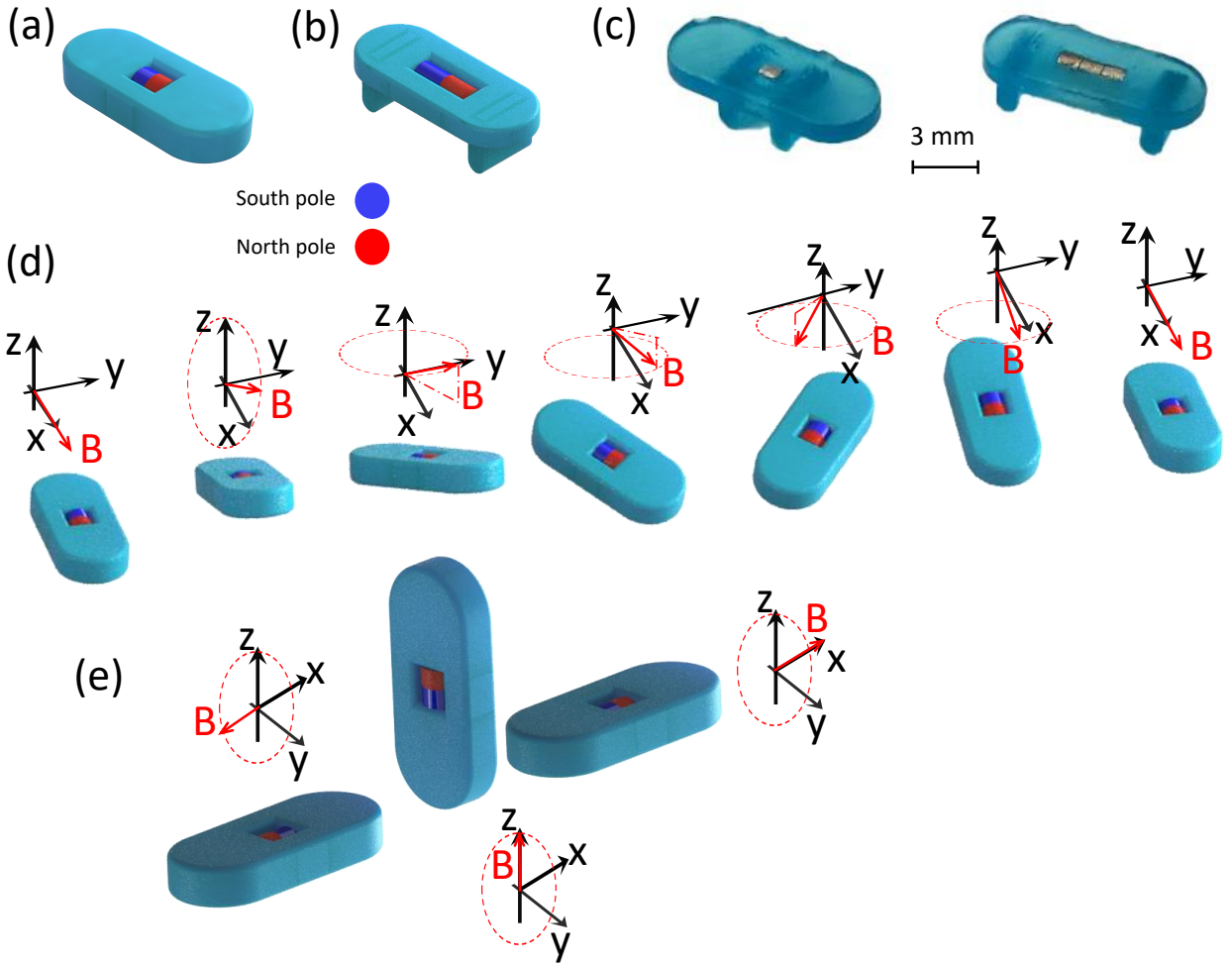


Figure 1.4: Millirobots and motion schemes of pivot walking and tumbling modes. **(a)** CAD design illustration of a millirobot without legs (primary design). The permanent magnet is embedded at the center of the body. **(b)** CAD design illustration of a millirobot with two legs (secondary design). Red and blue represent the north and south poles of a magnet. **(c)** Real 3D printed two millirobots with different pivot point separations. **(d)** Pivot walking motion; The scheme shows a complete locomotion step. It is achieved by lifting one end and forming a pivot point at the other end by rotating the magnetic vector around the y-axis; rotate the magnetic vector about the z-axis to turn the millirobot about the formed pivot; the process is then repeated in the opposite direction. **(e)** Tumbling motion; One tumbling step is achieved by rotating the magnetic vector about the y-axis by 180°.

applied to the system is eight amps and the system can generate a continuous magnetic fields above 10 millitesla (mT). We simulate the magnetic field profile at the center of the configuration using *Comsol* software as shown in Fig. 1.5(d).

1.6 Kinematic equations of motion

In the locomotion of the system, we assume that two tips of the body in the first design and legs in the second design are acting as the pivot points. A stationary electromagnet system produces a uniform rotating magnetic field in three dimensions. This rotating magnetic field generates torques on the magnets embedded into the millirobot. This aligns the long axis of the body with the applied magnetic field vector [2]. Thus, we are able to translate the center of mass of the body and perform rotations about the in-plane and out-of-plane angles.

1.6.1 Pivot Walking

Pivot walking is achieved by successively alternating the direction of the magnetic field vector in the positive and negative z-directions and rotating around z-axis as shown in Fig. 1.4(c). When the magnetic field vector is oriented in the positive z-direction, the induced magnetic torque presses one end down while the other end is lifted up. Subsequently, while having a pivot formed at the pressed end, a positive rotation about z-axis is applied. This causes the millirobot to rotate forward by a sweep angle of θ_i in the $x-y$ plane in its ith step. In the next step, the orientation of the magnetic field in z-direction is reversed, and the pivot moves to the other end. A negative rotation about z-axis is applied to rotate the millirobot by θ_{i+1} about the new pivot point. We consider this process as a complete step. Repeating this process, locomotion along a desired path is generated. Also, a single tumbling motion step is achieved by a rotation of the magnetic field vector about x-axis by a 180 deg as shown in Fig. 1.4(d).

Here, we calculate the coordinates of the center of mass (x_k, y_k), with k being the number of steps. The kinematic modeling of pivot walking depicted in Fig. 1.6(a) can be expressed as follows:

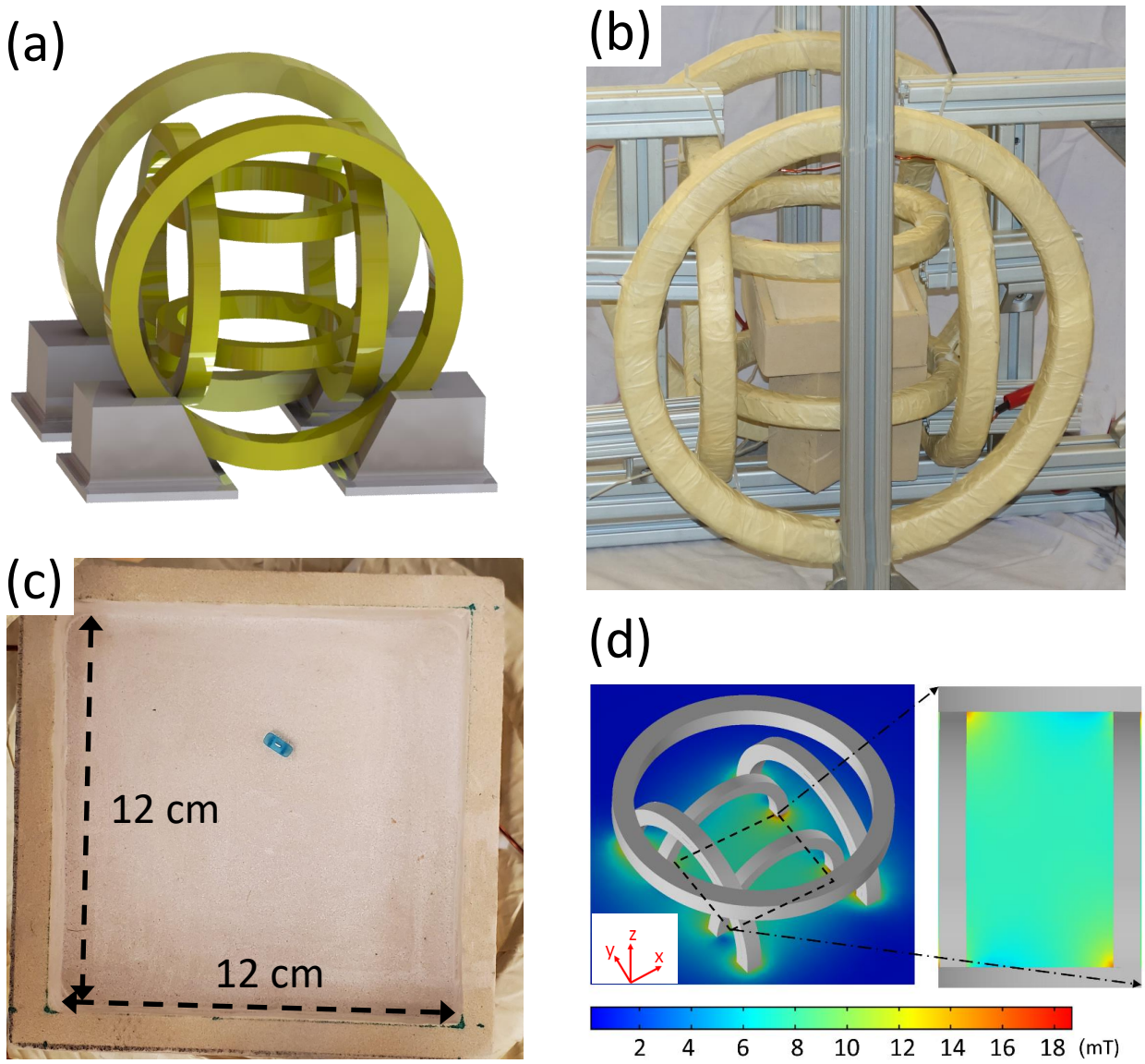


Figure 1.5: A nested Helmholtz electromagnetic coil system. **(a)** Isometric view of CAD design. **(b)** Isometric view of actual system. **(c)** Top view of workspace. **(d)** Simulation result of magnetic field at the center of the workspace.

$$x_k^n = x_0^n + \frac{L_n}{2}\sum_{i=1}^{k}\Bigg((-1)^i \cos\left[(-1)^i\left\lfloor\frac{i}{2}\right\rfloor\theta_1 + (-1)^{i-1}\left\lfloor\frac{i-1}{2}\right\rfloor\theta_2\right] + (-1)^{i-1}\cos\left[(-1)^{i-1}\left\lfloor\frac{i+1}{2}\right\rfloor\theta_1 + (-1)^i\left\lfloor\frac{i}{2}\right\rfloor\theta_2\right]\Bigg) \tag{1.15}$$

$$y_k^n = y_0^n + \frac{L_n}{2}\sum_{i=1}^{k}\Bigg((-1)^i \sin\left[(-1)^i\left\lfloor\frac{i}{2}\right\rfloor\theta_1 + (-1)^{i-1}\left\lfloor\frac{i-1}{2}\right\rfloor\theta_2\right] + (-1)^{i-1}\sin\left[(-1)^{i-1}\left\lfloor\frac{i+1}{2}\right\rfloor\theta_1 + (-1)^i\left\lfloor\frac{i}{2}\right\rfloor\theta_2\right]\Bigg) \tag{1.16}$$

where (x_0, y_0) are the coordinates of the initial position of the millirobot, n denotes the number of millirobot, L is the length, and (θ_1, θ_2) are the sweep angles around two pivot points, respectively. Also, the $\lfloor . \rfloor$ denotes the floor function, which is the function that takes as input a real number and gives as output the greatest integer less than or equal to the input.

1.6.2 Basic Motion Paths

One can obtain different motion paths by choosing different combinations of sweep angles. Three basic motion paths explored here as tools for swarm control; straight, triangular, and circular paths. Figure 1.6 shows the schematic representations of these paths. The straight-line motion shown in Fig. 1.6(b) is generated by choosing same sweep angles for each pivot step ($\theta_1 = \theta_2 = \theta_c$), however, the first sweep angle has to be the half of others (θ_1^1 or $\theta_1^2 = \theta_c/2$). The distance covered by millirobot, in this case, is proportional to the length of the millirobot.

A triangular trajectory is achieved by selecting equal sweep angles for the first kth steps ($\theta_{1:k}^1 = \theta_{1:k}^2 = \theta_c$) and the sweep angles are switched to negative θ_c. The two sets of steps are considered to be a complete step. If the millirobot starts from a line, after $2k$ steps (end by a complete step), it goes back to the same line (see Fig. 1.6(c)). The trajectory is an isosceles triangle, and equal base angles can be expressed in terms of the sweep angle ($\alpha = \frac{\pi - \theta_c}{2}$). The base ($d$) and height ($h$) of this triangle are related to the length of the millirobot, the

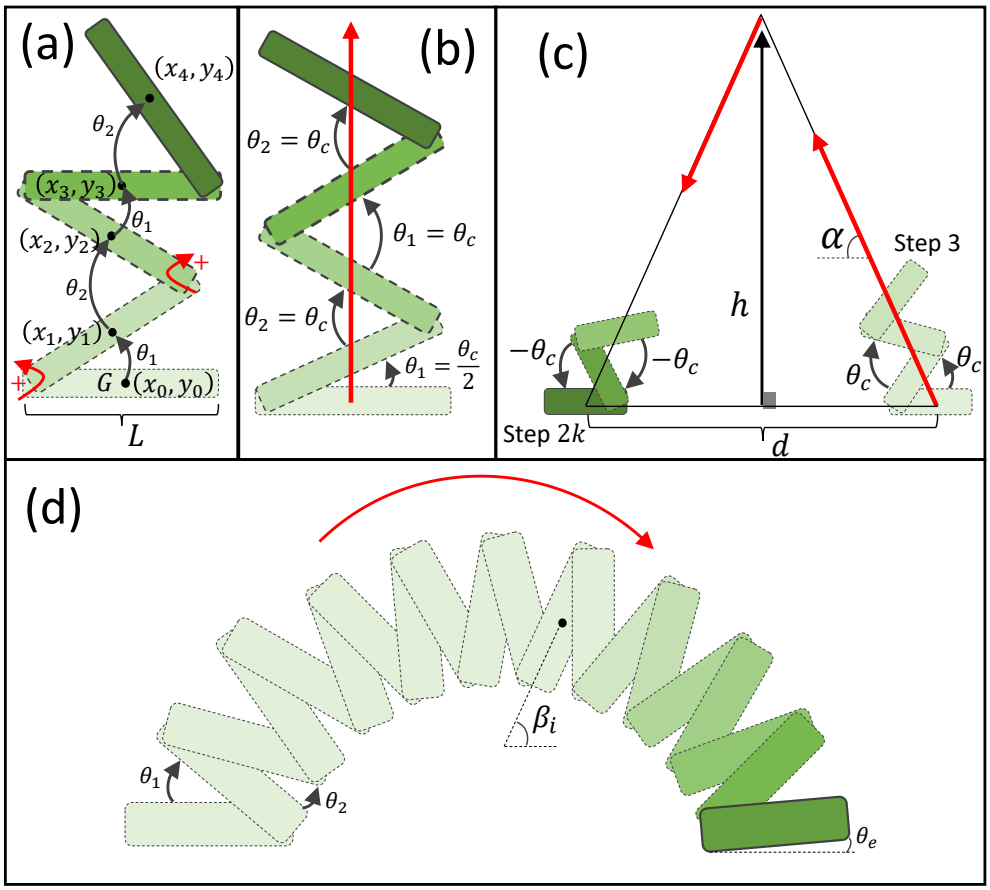


Figure 1.6: Millirobot walking tools **(a)** The schematic of the pivot walking with coordinates of the center of body and the positive directions of sweep angles. **(b)** Straight line motion. **(c)** Triangular trajectory path. **(d)** Circular path configuration.

sweep angle, and number of steps as follows:

$$h = \frac{d}{2}\cot\left(\frac{\theta_c}{2}\right) \tag{1.17}$$

$$y_k = h \tag{1.18}$$

In order to follow a circular path, two sweep angles must be different ($\theta_1 \neq \theta_2$). The radii of the generated circle is related to the sweep angles and length of millirobot. The equation of this trajectory can be found as:

$$x_i^2 + y_i^2 = r_c^2 \tag{1.19}$$

$$y_k = y_0 \tag{1.20}$$

where r_c denotes the radius of circular path (see Fig. 1.6(d)). If the millirobot starts from a line, after 2k-1 steps, it is not guaranteed to return to the same line. Thus, the sweep angle of the last step should be performed with a different sweep angle.

Consider a circular path with $2k$ steps. The millirobot moves k steps with sweep angle θ_1 on the first pivot point and $k-1$ steps with sweep angle θ_2 on the second pivot point. The angle between the long axis of the body and the positive direction of x-axis in each step (β_i) and the extra sweep angle (θ_e) to complete the round (last step or $2k^{th}$ step), can be calculated as:

$$\theta_d = \left\lfloor \frac{i+1}{2} \right\rfloor \theta_1 - \left\lfloor \frac{i}{2} \right\rfloor \theta_2 \tag{1.21}$$

$$\beta_i = \begin{cases} \theta_d & \theta_d \leq 90^\circ \\ 180^\circ - \theta_d & \theta_d > 90^\circ \end{cases} \tag{1.22}$$

$$\theta_e = 180^\circ - (k\,\theta_1 - (k-1)\theta_2) \tag{1.23}$$

We utilize these trajectories as tools to conduct swarm positioning control.

1.6.3 Swarm Motion Using Basic Motion Paths

Here, we consider two millirobots with different pivot separations. Initially they are placed on a straight line with a separation of Δr between them. By using triangular path planning, one can change the distance between the millirobots and reverse their initial order on the original line (see Fig. 1.7).

To express the position alteration of two millirobots more accurately, we conduct a parametric analysis of the effects of sweep angle and number of steps on the final positions (see Fig. 1.8). Figure 1.8(a) depicts the effect of the total number of steps on the final distance between two millirobots when a constant sweep angle $\theta_c = 24^\circ$ is used. A negative value for distance means the order of two millirobots is preserved. Also, Fig. 1.8(b) shows the alteration in the relative position of millirobots in terms of changing sweep angles in a constant number of steps $2k + 1 = 33$. The direction of the path is altered at $k = 12$. In Fig. 1.8(c), one can see the difference in distance between two millirobots at the end of the triangular path motion as a function of the sweep angle and number of steps (see **SP** 2).

Based on the results shown in Fig. 1.8, we can claim that with a specific combination of sweep angle and number of steps, any two millirobots with different lengths can walk to final positions with their relative distance set to an arbitrary desired value. Next, we present a formal mathematical formulation for this type of action; consider two millirobots with different lengths L_1 and L_2 starting on a line with a relative distance of Δr. Whereas, Δp denotes the final desired value of the relative distance between two millirobots. From Eq. (1.15), the base of the triangle can be found as:

$$d = |x_k^n - x_0^n| = f_x(L_n, \theta_c, k) \tag{1.24}$$

where $f_x(.)$ is a function of three parameters (L_n, θ_c, k), which can be extracted from the right hand side of Eq. (1.15) and the combination set can be expressed as follows:

$$S = \{(\underset{i=1,2}{L_i}, \theta_c, k) \mid d_i = \underset{i=1,2}{f_x}(L_i, \theta_c, k), d_2 = \Delta r + d_1 \pm \Delta p\} \tag{1.25}$$

where d_1 and d_2 are the bases of triangular paths for each millirobot. One can use this

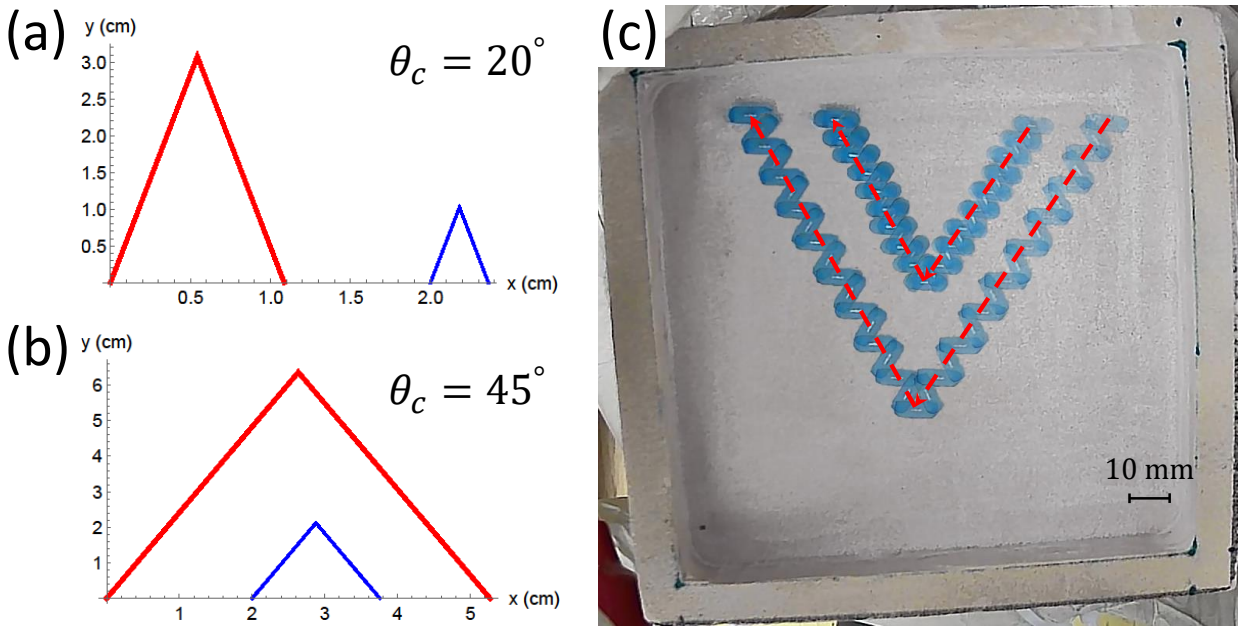


Figure 1.7: Changing the final distance between two millirobots. The initial distance between two millirobots is $\Delta r = 2\,\text{cm}$ and the lengths of the robots are 1.5 cm for red trajectory and 0.5 cm for blue one. In both paths, the millirobots change the direction after 8 steps. **(a)** The sweep angle is $\theta_c = 20^\circ$ and the distance after at the end is approximately 1.25 cm. **(b)** The sweep angle is $\theta_c = 45^\circ$ and the distance at the end is approximately 1.4 cm, but the order of the millirobots is changed. **(c)** Experimental results of the position altering of two millirobots. The pivot separations are 5 and 9 mm.

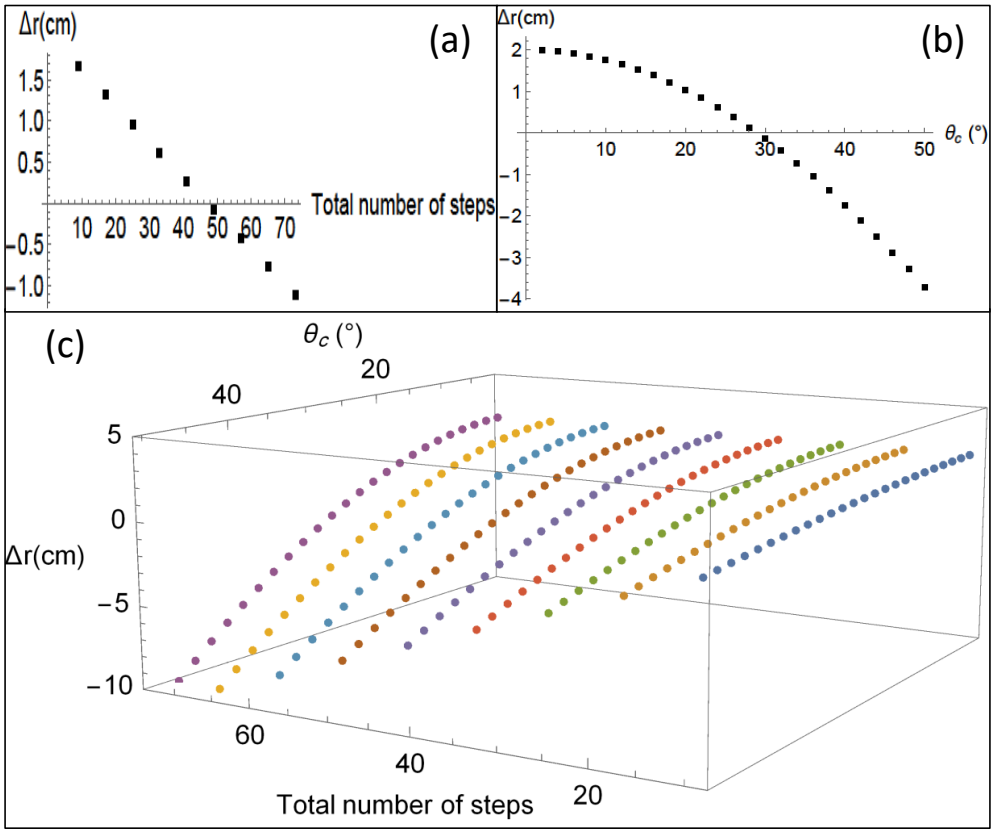


Figure 1.8: Variation of the final distance between two millirobots with lengths of 2 and 1 cm for different sweep angles and number of steps, when the initial distance is 2 cm. **(a)** The sweep angle is fixed at $\theta_c = 24°$ and the number of steps is varied. **(b)** The sweep angle is varied when the number of steps sets at 33. **(c)** Varying both sweep angle and number of steps.

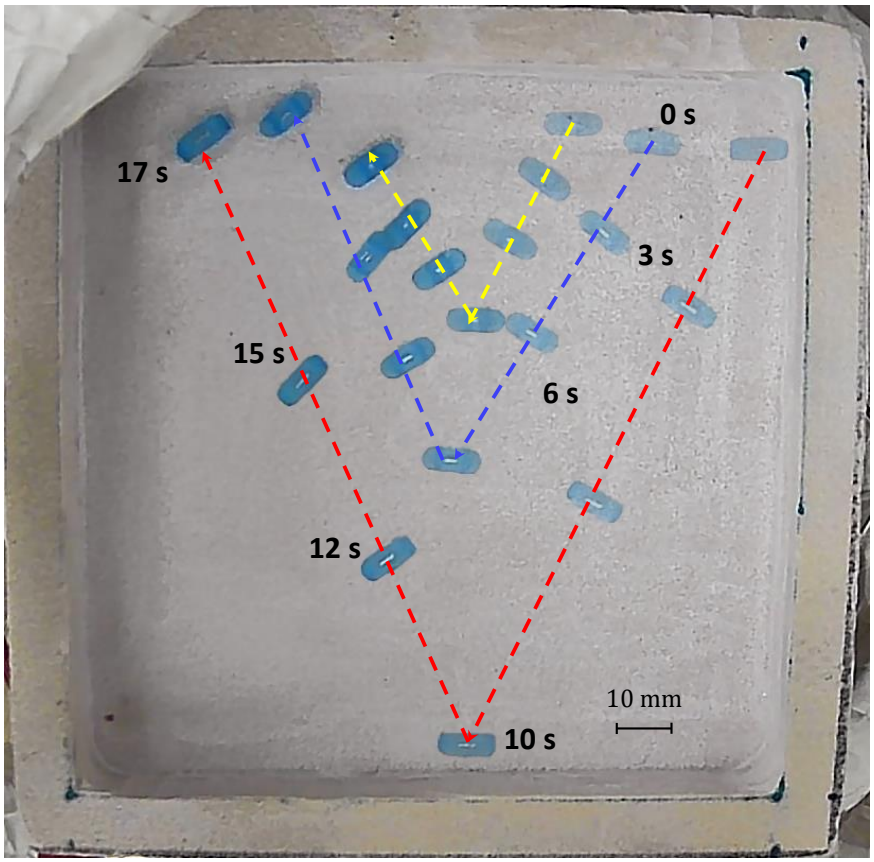


Figure 1.9: Sequences of changing order of three millirobots with 3, 5, and 9 mm in pivot point separations.

motion path to change the order of any number of millirobots initially placed on a straight line. Subsequently, we conduct an experiment to show this ability only for three millirobots with different pivot separations due to the restriction imposed by the size of the workspace (see Fig. 1.9). We should note that in the following figures of the experimental results, we just show a select number of steps in the overlay pictures to highlight the overall path of the swarm motions without overcrowding the figures. One can see the experiments in the videos provided in the supplementary materials.

We present a swarm motion of two millirobots using the basic paths. This swarm motion shows the capability of the walking tools. We assume that the initial and final positions of two millirobots are (r_1, r_2) and (p_1, p_2) respectively. Without loss of generality, we consider that the length of millirobot 2 (m_2) is greater than millirobot 1 (m_1) $(L_1 < L_2)$. Thus, m_2

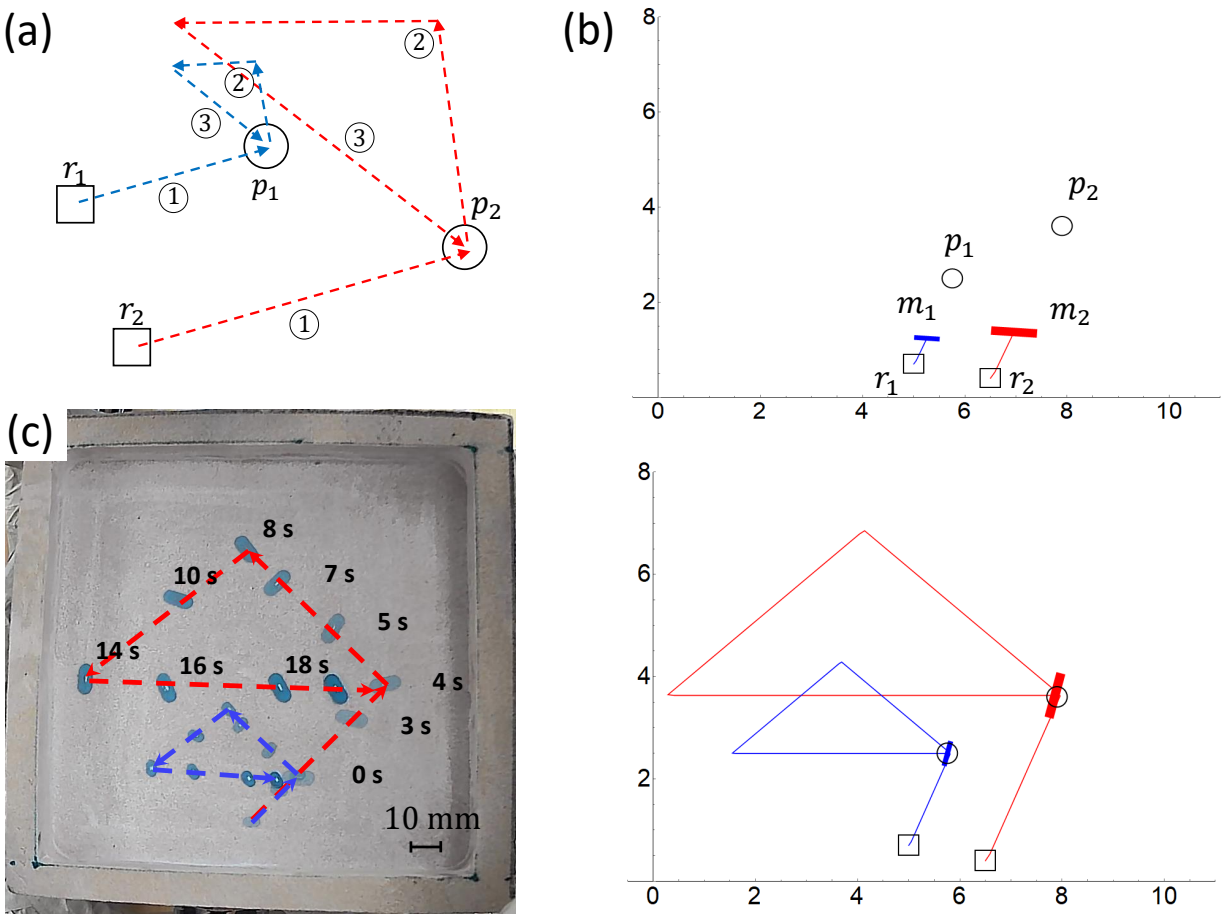


Figure 1.10: Swarm pattern motion of two millirobots with 3 and 9 mm lengths. **(a)** Illustration of swarm motion. Initial positions are marked by squares and final ones by circles. Dashed lines show the paths of the centers of millirobots in three runs. **(b)** Simulation result of walking two millirobots. **(c)** The experimental result of the swarm motion of two millirobots.

moves faster and undergoes longer triangular path. This motion consists of three walking runs including a line with a slope, triangular path, and straight line motions to generate a swarm pattern motion. Figure 1.10 shows an illustration of simulation and experimental results of the swarm motion of two millirobots with different pivot separations.

1.7 Swarm Position Control of n Millirobots

The proposed swarm control methodology often requires a priory determination of the lengths of the millirobots required to perform a specific placement task. In this section, we present an algorithm that yields the required robot lengths, sweep angles, and number of

Algorithm 1: Swarm control-n millirobots

Input parameters: The coordinates of initial positions $(x_0^i, y_0^i\,; i = 1:n)$ and final destinations $(x_f^i, y_f^i\,; i = 1:n)$.

Calculation:

1: Set length of m_1 as L_1 and move to its final position p_1

2: From Eqs. (1) and (2), solve for first set:
$S_1 = \{(\theta_{c1}, k_1)\,|\,\text{Straight motion}\,,\,(x_0^1, y_0^1) \rightarrow (x_f^1, y_f^1)\}$

3: Use set S_1, and calculate the lengths of other millirobots:
$S_i = \{(L_i)\,|\,\theta_1 = \theta_2 = \pm\theta_{c1}\,,\,(x_0^i, y_0^i) \rightarrow (x_f^i, y_f^i), i = 2:n\}$

6: Swarm set: $S_{swarm} = \bigcup_{i=1}^{n} S_i$

Figure 1.11: Swarm control algorithm

steps to move millirobots from their initial positions to desired final destinations.

Algorithm 1 is used to position n millirobots from initial positions to final desired ones. This positional control of n millirobots can be carried out by calculating the lengths of millirobots based on their initial positions and desired final destinations. The algorithm presents the process of finding different sets of (L_n, θ_n, k_n) to perform this task.

In order to position n millirobots at their corresponding final destinations, one can use Algorithm 1.11. Let's consider the first millirobot (m_1) as it moves to its final position with straight motion based on Eqs. (1.15) and (1.16), by calculating the set $S_1 = \{(\theta_{c1}, k_1)\,|\,\text{straight motion}\,,\,(x_0^1, y_0^1) \rightarrow (x_f^1, y_f^1)\}$. Using this set and applying it to Eqs. (1.15) and (1.16) for other millirobots, the coordinates of midpoint position of each millirobot $(x_1^i, y_1^i\,; i = 2:n)$ can be expressed as a function of their lengths $(L_i\,; i = 2:n)$, sweep angle, and number of steps. The relative distance between the initial and final position of

a midpoint of each millirobot is also a function of its length, sweep angle, and number of steps. By solving the resulting equations, one can get the sets of millirobot lengths, sweep angles, and the number of steps off-line. Then, the millirobots can be placed at their initial positions and the swarm control can be conducted by applying the solution sets.

1.8 Controllability Analysis

The kinematic equations of motion of each robot can be represented using the unicycle model as follows.

$$\dot{x} = x + u_r \cos(\theta + \frac{\pi}{2}) + u_p \cos\theta \tag{1.26}$$

$$\dot{y} = y + u_r \sin(\theta + \frac{\pi}{2}) + u_p \sin\theta \tag{1.27}$$

or in matrix form:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u} \tag{1.28}$$

where

$$\bar{x} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.29}$$

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{1.30}$$

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{1.31}$$

$$\bar{u} = \begin{bmatrix} u_r \cos(\theta + \frac{\pi}{2}) \\ u_r \sin(\theta + \frac{\pi}{2}) \\ u_p \cos\theta \\ u_p \sin\theta \end{bmatrix} \tag{1.32}$$

$$\dot{X} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} X + \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -u_r \sin(\theta) \\ u_r \cos(\theta) \\ u_p \cos\theta \\ u_p \sin\theta \end{bmatrix} \tag{1.33}$$

where u_r is the rolling speed and u_p is the walking speed.

This model can be generalized to describe the swarm robots system as follows.

$$\dot{\bar{\mathbf{x}}} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{B}\bar{\mathbf{u}} \tag{1.34}$$

where $\mathbf{A}$ is $2n \times 2n$ identity matrix, $\mathbf{B}$ is $2n \times 4$.

$$\bar{\mathbf{x}} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix}_{2n\times 1} \tag{1.35}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}_{2n\times 2n} \tag{1.36}$$

$$\mathbf{B} = \begin{bmatrix} -1 & 0 & v_1 & 0 \\ 0 & 1 & 0 & v_1 \\ \vdots & \vdots & \vdots & \vdots \\ -1 & 0 & v_n & 0 \\ 0 & 1 & 0 & v_n \end{bmatrix}_{2n\times 4} \tag{1.37}$$

$$\bar{\mathbf{u}} = \begin{bmatrix} u_r \sin(\theta) \\ u_r \cos(\theta) \\ u_r \sin(\theta) \\ u_r \cos(\theta) \end{bmatrix} \tag{1.38}$$

where

The controllibility $\mathbb{C}$ of the swarm system can formulated as follows.

$$\mathbb{C} = \left[B, AB, A^2B, \cdots, A^{2n-1}B \right] \tag{1.39}$$

For one robot, the matrices $\mathbf{A}$ and $\mathbf{B}$ are the same as matrices on Eqs. (1.30) and (1.31); then, the rank of $\mathbb{C}$ in this case will be two, which means all degrees of freedom (DOFs) are controllable. For two robots, the matrices $\mathbf{A}$ and $\mathbf{B}$ are represented as:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.40}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & v_1 & 0 \\ 0 & 1 & 0 & v_1 \\ 1 & 0 & v_2 & 0 \\ 0 & 1 & 0 & v_2 \end{bmatrix} \tag{1.41}$$

Here, the rank of $\mathbb{C}$ will be 4, which means all DOFs are controllable. This swarm control is shown in Fig. 1.10. In this experiment, the robots are controlled to place to their corresponding desired final positions. Also, in the parametric analysis done in Fig. 1.8, it is shown that the relative distance between two robots can get any values. It means they can be steered to any desired position with a certain sequence of pivot walking and tumbling modes motion.

The matrices $\mathbf{A}$ and $\mathbf{B}$ for three robots can be written as:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.42}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & v_1 & 0 \\ 0 & 1 & 0 & v_1 \\ 1 & 0 & v_2 & 0 \\ 0 & 1 & 0 & v_2 \\ 1 & 0 & v_3 & 0 \\ 0 & 1 & 0 & v_3 \end{bmatrix} \tag{1.43}$$

and the rank of $\mathbb{C}$ in this case will be 4, which means only 4 DOFs are controllable as we have seen in the previous case. Therefore, it is shown that only two robots can be controlled.

The controllability analysis shows that swarm systems with more than four DOFs are not controllable. This is because the rolling input moves all robots by applying the same rigid-body transformation. Similarly, the pivot walking also applies the same rigid body transformation, but scaled by the constant v_i for each robot. This constant linearly depends on the length of the robot.

1.9 Results

1.9.1 Swarm Pattern Motions

In this section, we experimentally demonstrate the swarm position control of the millirobots to generate different geometrical shapes including triangle, square, pentagon (see Fig. 1.12), and hexagon patterns (see Fig. 1.13). The edges of each shape are considered as the final desired positions of each millirobot. In these experiments, we use the secondary design of the millirobots, in which the pivot separations are fixed at 3, 5, 7, and 9 mm.

By using the reverse solutions of Algorithm 1, one can find the desired initial positions to perform a swarm motion. In addition, we use the tumbling motion to move the final shape of patterns. This capability is the strength of the presented swarm motion. Any desired pattern can be generated through Algorithm 1 and then the final pattern can be placed anywhere by using the tumbling mode motions. Figure 1.13 depicts the experimental result of the swarm position motions to generate a Hexagon pattern. This swarm motion is conducted by six millirobots with three different lengths. We use two millirobots of each length, which are 3, 7, and 9 mm. The millirobots start at points $r_i;\ i = 1:6$. They move to points $p_i;\ i = 1:6$ in pivot walking mode to generate the desired hexagon pattern. Then, they are steered in tumbling mode to reach their corresponding final points ($q_i;\ i = 1:6$). Also, it should be noted that in the figure we only show the experimental results of the hexagon pattern to reduce the complexity.

1.9.2 Swarm application

The objective of swarm control of a group of robots is to perform a task, which is not possible to perform with a single robot. Here, we numerically and experimentally show practical applications of the swarm motions of these millirobots. We should note that we assume that there are no collisions among the millirobots during their motions.

1.9.2.1 Expansion maneuver

A group of four primary-design millirobots conducts a maneuver that expands from a contracted initial formation to an expanded one. We call this maneuver "Expansion". The simulation of the expansion maneuver is conducted as follows; the group starts from a relatively compact of initial positions. Then, a circular motion is performed to place the robots in front of a narrow opening while fixing their relative distances at specific values. This formation makes it possible for the group to go through the channel by using straight-line motion. After passing the channel, depending on whether we require horizontal or vertical expansion, the subsequent scenarios are different. For instance, after formation undergoes the walking sequences to pass through the channel, they are steered on an inclined straight line to adjust their positions on subsequent circular paths. In the end, a circular motion is carried out to bring all millirobots to their final positions. (see Fig. 1.14 (a)). We experimentally

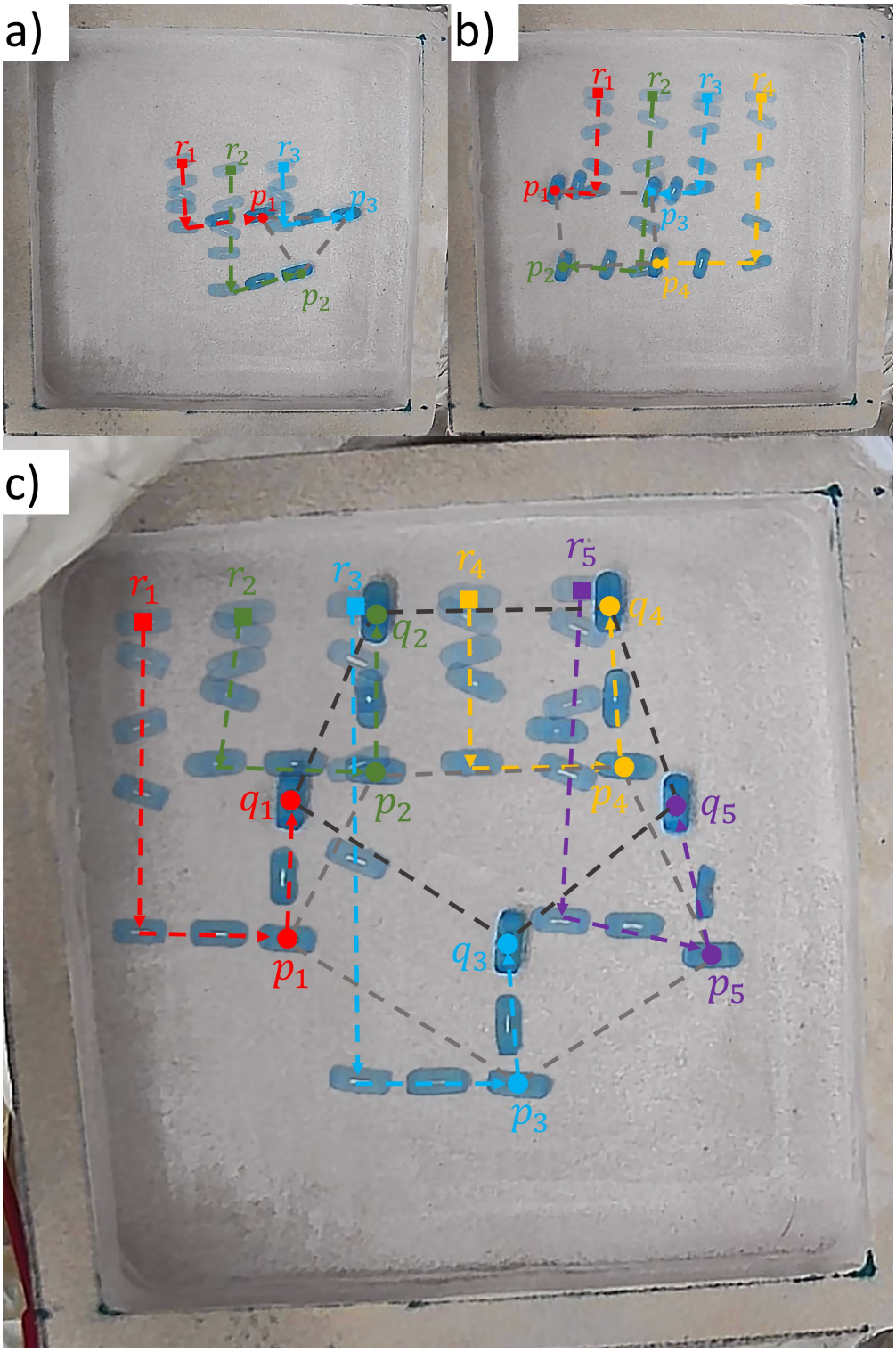


Figure 1.12: Swarm different patterns formation. a) Triangle. b)Square. c) Pentagon. Dashed color-line represents the path of the midpoint of each millirobot. The initial (r_is) and final (p_i) positions are shown by squares and circles symbols. The second final positions (q_is) are shown the preserved patteren after tumbling mode motion.

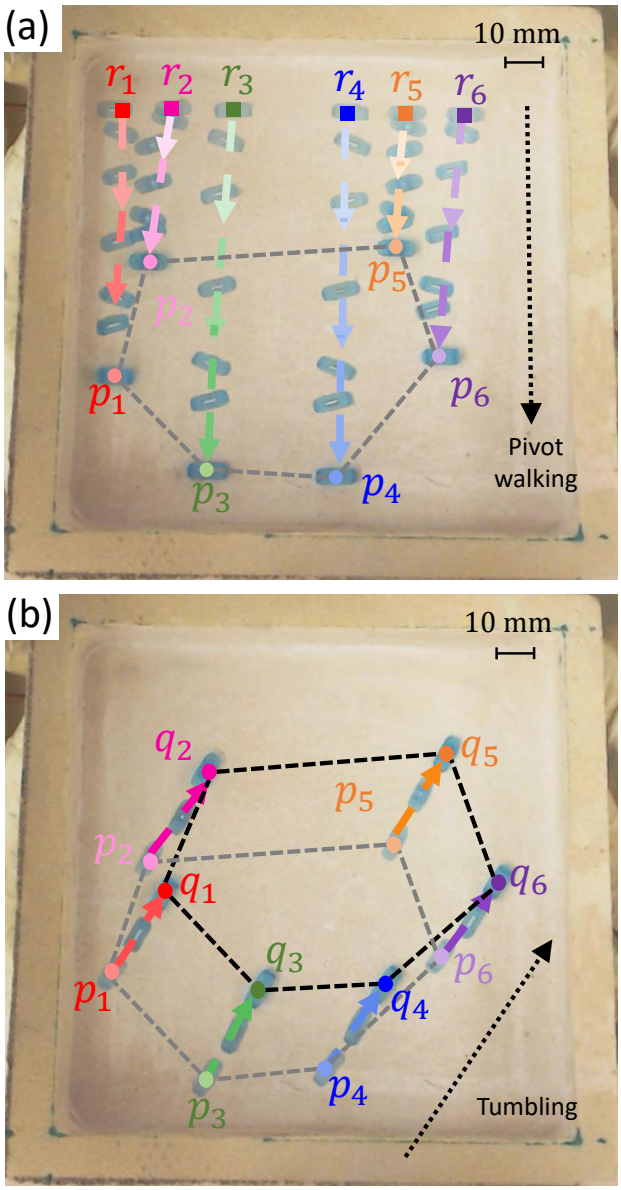


Figure 1.13: Swarm motions of six millirobots to generate a hexagon pattern. Each dashed color-line represents the path of the midpoint of each millirobot. The initial and final positions are shown by squares and circles symbols. **(a)** Six millirobots start from their initial positions (r_i; $i = 1:6$) and move to points p_i; $i = 1:6$ in pivot walking mode to generate a hexagon pattern. **(b)** The millirobots are steered to their final positions (q_i; $i = 1:6$) in tumbling mode while the hexagon pattern is preserved.

demonstrate the expansion maneuver conducted by four millirobots. They are placed in specific compact initial positions to minimize the attraction forces between the magnets. Figure 1.14(b) shows a selected sequence of frames of experimental results of the expansion maneuver. The millirobots start from their initial positions ($r_i;\ \ i = 1:4$). They move in a circular path to a location in front of the opening; then, walk in a straight line formation to go through the channel. In the end, circular and line motions are performed to expand the formation and reach final destinations ($p_i;\ \ i = 1:4$).

1.9.2.2 Contraction maneuver

A group of millirobots is placed on the respective expanded formation of initial positions. Then, they undergo a reverse sequence of motions explained in the expansion maneuver. They are steered to the intermediate compact positions in front of a narrow channel and walked through it. In the end, a maneuver is performed in order to place the millirobots in respective desired final positions. Alternatively, the contraction maneuver can be named as the reverse maneuver of expansion motions.

1.9.2.3 Reverse maneuver

Under the same control input, all millirobots move in the same direction. A challenging task is to steer the millirobots in opposite direction by applying the same magnetic field. We propose a walking plan to move two millirobots from a set of initial positions to the desired set of final destinations, which is required an opposite direction motion. We name this maneuver "Reverse". Figure 1.15(a) shows an illustration of this maneuver. This walking plan consists of five sequences; including two pivot walking and three tumbling mode motions. The millirobots are placed on the top side of the two obstacles. They should pass a narrow channel with a width of W_c and be positioned at the other side of the obstacles.

We conduct an experiment to show the reverse maneuver as follows (see Fig. 1.15(b)); two millirobots with different pivot separations (3 and 9 mm) are placed at their corresponding initial positions (r_1, r_2). Here, the millirobot placed at r_1 has bigger pivot separation than the other one. First, they move to the point 1 in pivot walking mode to reduce the relative distance between them until it reaches less than W_c. Then, they tumble back to the point 2, as the relative distance between the millirobots remains constant. They tumble through

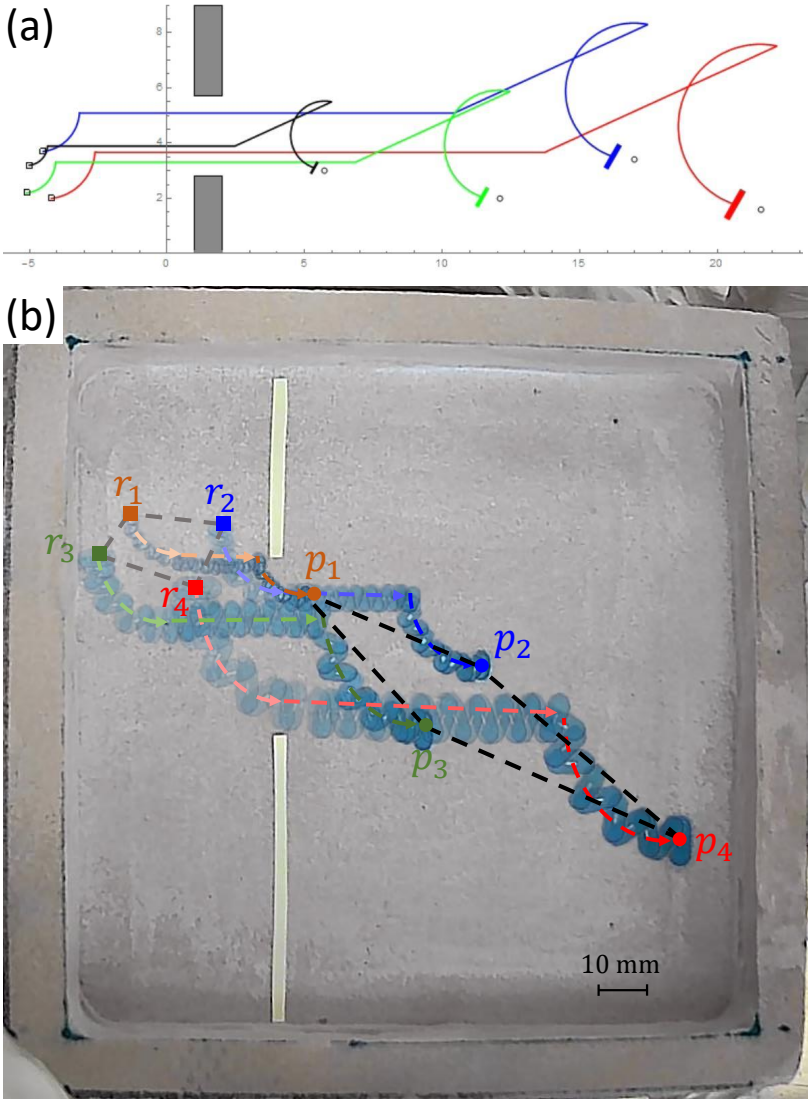


Figure 1.14: Sequences of the horizontal expansion maneuver of four millirobots with different lengths as 3, 5, 7, and 9 mm. The compacted initial and the final expanded formations are indicated by gray and black dashed polygons respectively. Each dashed line represents the path of the midpoint of each millirobot.

the channel and pass it to the point 3. Then, they are steered to the point 4 in the pivot walking mode to increase the relative distance between them. At the end, they tumble to their final destinations (p_1, p_2).

We should note that the experiments are conducted on a dry surface. Thus, we observe slippages at the pivot points in different situations. As we have pre-defined control inputs acting on the system, we can not overcome the effects of the slippages on the final outputs. This causes the experimental misplaced positions of millirobots from their simulated and desired final destinations. There are not any trends or similarities for the misplacements and are due to un-modeled friction on the surface. The solution for this problem would be to add a closed-loop controller to correct the motion of each millirobot, which will be proposed in the next chapter.

1.10 Conclusion

This study proposed a pre-computational technique for swarm position control of a group of small-scale robots using uniform input. We presented an algorithm for the positioning n-robots actuated by similar forces. The unique millirobots introduced in our previous study [2] modified in design. We placed magnets at the center of the body to reduce the magnetic attraction forces. Also, we added two legs acting as pivot points. In new design, by varying the pivot separation and keeping identical lengths, the millirobots can move in different velocities in pivot walking mode and constant velocity in tumbling mode. To obtain different positional outcomes out of steering millirobots under the same control input, we used millirobots with different lengths as well as variable pivot separation in pivot walking mode.

First, we presented two modified designs of millirobots and itemized their advantages. Then, we demonstrated different walking tools, which were utilized for the swarm motions. We designed an algorithm to place n millirobots from arbitrary initial positions to other arbitrary final positions. The lengths of millirobots were calculated based on the coordinates of the initial and final positions and a pre-computational path planning to perform the swarm motions. We verified the proposed algorithm for swarm positioning the millirobots through simulation and experiments. Also, we conducted different experiments to show the capability

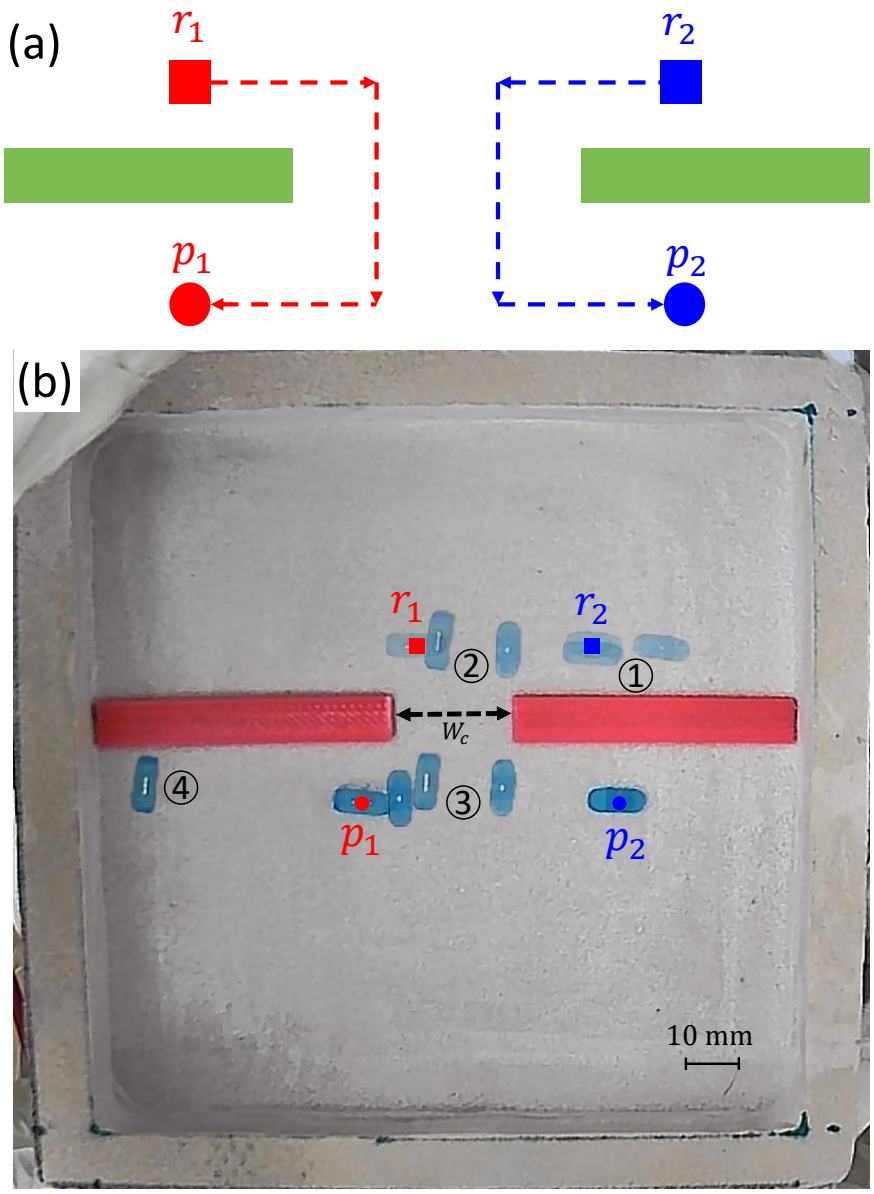


Figure 1.15: Sequences of the reverse maneuver of two millirobots with different pivot separations as 3 and 9 mm.

of our millirobots to perform a group task.

In the experiments, we faced the misplaced conditions for millirobots, which was the result of slippage from lack of necessary friction at pivot points. For the next step, we are working on a closed-loop control strategy to conduct more precise experiments.

Acknowledgments

We would like to thank Mr. Necdat Yildimier for his help in fabricating the magnetic coils and its holding frame and also, Mr. Mohammad Karim Dehghan Manshadi for his guidance in preparing the *Comsol* simulations.

Feedback Control of Millimeter Scale Pivot Walkers Using Magnetic Actuation

1.11 Problem description

An external magnetic field can be used to remotely control small-scaled robots, making them promising candidates for diverse biomedical and engineering applications. In this section, we introduce a magnetically actuated robot that navigates in a horizontal plane. Previously, we showed that our proposed robot is highly agile and can perform a variety of locomotive tasks such as pivot walking, tumbling, and tapping. Here, we focus on the closed-loop controlling of the locomotion of the millirobot in the pivot walking mode. The role of the sweep and tilt angles in robot's motion is also investigated.

We propose two controllers to regulate the gait of the pivot walker. The first one is a proportional-geometric controller, which determines the correct pivot point that the millirobot should use. Then, it regulates the angular velocity proportionally based on the error between the center of the millirobot and the reference trajectory. The second controller is based on a gradient descent optimization technique, which expresses the control action as an optimization problem. These control algorithms enable the millirobot to generate stable gait while tracking a desired trajectory. We conduct a set of different experiments and simulation runs to establish the effectiveness and the robustness of proposed controllers for different sweep angles. Robustness is measured in terms of the tracking error. The two controllers exhibit excellent performance, but it is observed that gradient descent based controller yields faster convergence time, smaller tracking error, and fewer number of steps. Finally. We perform a vast experimentally parametric analysis on the effect of the sweep angle, tilt angle, and step time on the tracking error. As we expect, the optimization-based controller outperforms the geometric based controller.

1.12 Introduction

Biologically inspired locomotion and the growth in need for mobility in challenging and unknown environments have motivated many researchers to design different miniature robots. These robots can be listed as conventional wheeled, crawling and snake-like, legged, and hop-

per. Recently, traversability in irregular environments has become an important feature in the design of robots. Small-scale robots show great potential capabilities in maneuverability and reachability in such environments. Also, these robots can benefit from untethered actuation mechanisms and free from limitations imposed by onboard actuators and power sources. Untethered tiny robots can be actuated by external forces, such as electromagnetic fields [21,32], acoustic waves [33], and light [34–36].

Small scaled robots actuated and controlled by external magnetic fields can be used in different applications such as accessing and manipulating objects in tight spaces. Various locomotion modes has been developed for these types of robots such as *Pulling/Pushing* [37], *Tumbling/Rolling* [38], *Helical Thrusting* [39], *Swimming* [40], *Crawling* [41], *Stick-Slipping* [42], and *Pivot Walking* [2]. These locomotions can be carried out by rigid or soft mechanisms [43].

Here, we summarize these different motion modes. Steering a permanent magnet can generate a magnetic force and subsequently pull or push a magnetic body on a dry surface or in a fluid environment. Diller *et al.* [24] controlled a sub-millimeter robots by using a magnetic gradient pulling method. The resulting tumbling motion exhibited a more controllable mode without significant slippage. Bi *et al.* [44] presented a microrobot that is controlled by an external magnetic field. A permanent magnet was rotated beneath the workspace and regulated a microrobot to tumble and travel through a narrow channel. Alternatively, when a rotating magnetic field is applied to a helical robot submerged in a fluid, it will move in the direction of the helical axis. Bozuyuk *et al.* [39] 3D printed a double-helical microswimmer made from a magnetic polymer composite. The motion of this robot was controlled using light in order to perform drug payload carry and release tasks.

In addition, robots can swim in a liquid medium by producing a wavy motion of their bodies. Zhang and Diller [45] designed a flexible magnetic sheet, which can swim in a liquid. Using the same technique, one can also generate crawling motion on dry surfaces. Accordingly, Hu *et al.* [41] developed a non-uniform magnetized soft millimeter-scale robot and controlled it to move on a solid surface, swim inside and on the surface of a liquid, and crawl in a channel.

Using magnetic actuation makes it possible to significantly reduce the size of pivot walkers

because the burden of having on-board actuators is now moved elsewhere. This type of magnetically actuated locomotion was used by Dong and Sitti [46] to regulate the motion of a microgripper. They controlled the gripper to reach and grip an object, then move it and release it at the desired destination.

In this section, we propose two control algorithms that track time-dependent desired trajectories based on millirobot's kinematics. We verify our proposed controllers through simulation and experimental results.

The first scheme is a proportional-geometric controller, which moves the center of the millirobot to the desired location at that time instant. The controller performs this task by taking successive steps alternating between the two pivot points. During each step, based on the distances between the pivot points and the desired position, one of the pivot points is chosen. Then, the millirobot swings about the chosen pivot with a velocity that is proportional to the distance between the center of the millirobot and the desired point.

The second control scheme is based on a gradient descent optimization technique, where the control problem is expressed as an optimization problem. The control inputs are the parameters that need to be optimized to reduce the error between the center of the millirobot and the desired trajectory. These control algorithms enable the millirobot to walk and track an arbitrary trajectory.

The locomotion of the millirobot depends on three parameters, sweep angle, tilt angle, and generating path step time. We conduct extensive experiments to perform parametric analysis on the effects of varying these parameters on the trajectory tracking error.

1.13 Kinematic model

As we discussed before, the magnetic millirobot is actuated by an external magnetic field $\mathbf{B}$. The induced magnetic torque will rotate and align the millirobot's permanent magnetic $\mathbf{M}$ with the external magnetic field. As shown in Fig. 1.17, the pivot points are labeled as $\boldsymbol{A}$ and $\boldsymbol{B}$. A pivot point is formed when one end is pressed down while the other end is lifted up by a tilt angle (α) by applying the induced magnetic field in that direction.

Derivation of the kinematics of the pivot walker is based on the assumption that the millirobot is constructed as a rigid body. To identify the millirobot position on the plane,

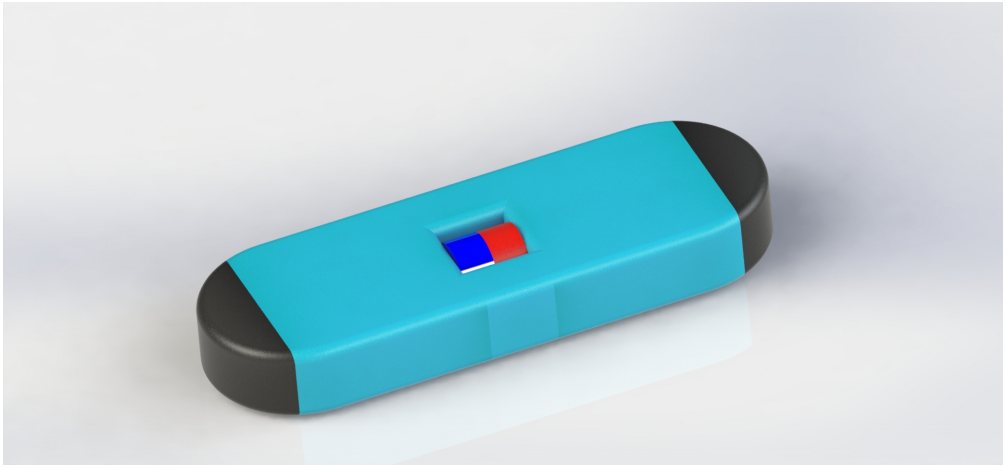


Figure 1.16: The CAD design of a pivot walking millirobot. A permanent magnet is embedded at the center. The red and blue sides are represented the north and south magnetic poles, respectively.

two reference frames are used; the global and the local reference frames as shown in Fig. 1.17. The axes X_I and Y_I define an arbitrary inertial basis in the global reference frame. To specify the position of the millirobot, a point C on the center of the millirobot is chosen. The basis $\{x_C, y_C\}$ defines the millirobot's local reference frame that passes through C. The position of C in the global reference frame is specified by coordinates x and y, and the angle between the two frames is given by θ. The robot position is then fully identified by the three variables x, y, and θ. A 3×1-vector q_I is defined to describe the millirobot state in the global frame and q_R in the local frame.

$$q = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \tag{1.44}$$

The transformation between the millirobot frame and inertial frame is described as follows:

$$\dot{q}_R = R(\theta)\dot{q}_I \tag{1.45}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.46}$$

where $R(\theta)$ is the rotation matrix. Based on Eq. (1.45) the kinematics of the millirobot in

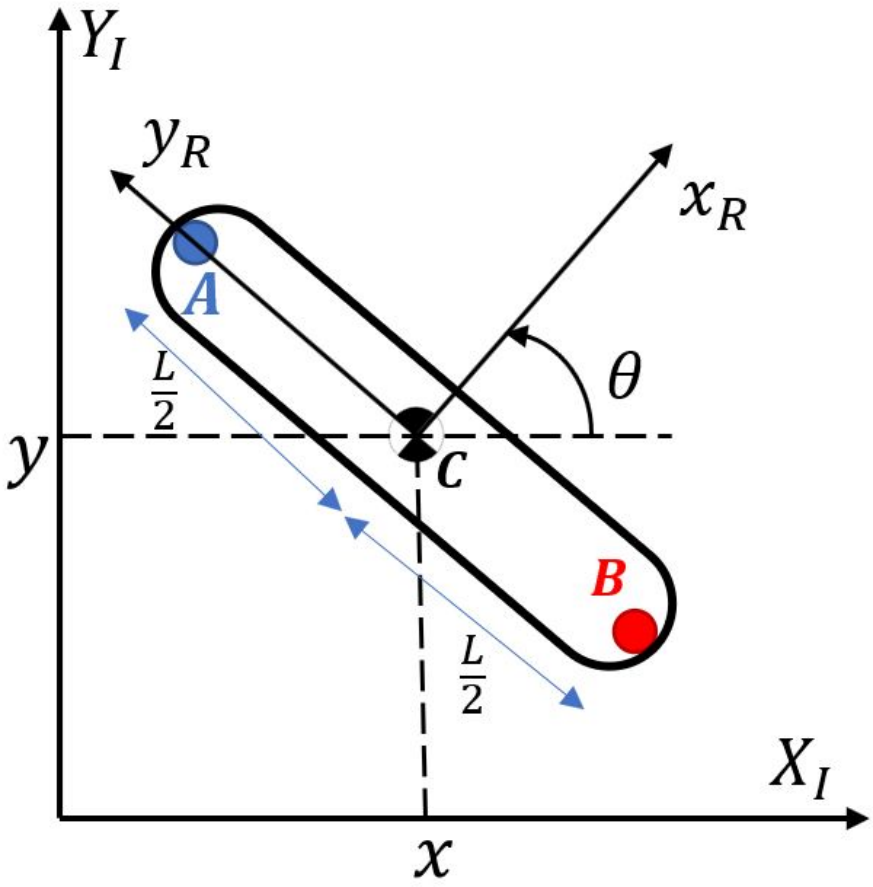


Figure 1.17: The global reference frame and the millirobot local reference frame.

the inertial frame can be written as follows:

$$\dot{q}_I = R(\theta)^{-1}\dot{q}_R \tag{1.47}$$

Thus, we need to obtain the kinematic equations in the millirobot frame. But, before presenting the derivation of the kinematic model of the pivot walker, an assumption should be presented. The robot can only rotate about the pivots without slippage. Figure 1.18 describes how the millirobot is rotating about each pivot, wherein Fig. 1.18(a), the active pivot is A, therefore this point is fixed and the millirobot is rotating about it. While point B is fixed when pivot B is active as shown in Fig. 1.18(b). Successive switching between the two pivots will enable the millirobot to generate locomotion.

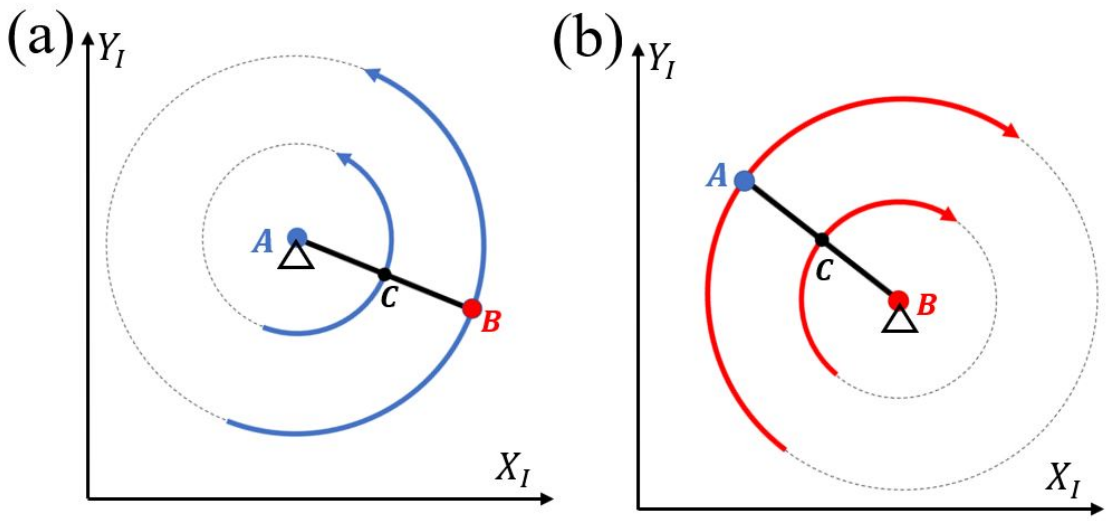


Figure 1.18: The robot motion about each pivot.

Since the millirobot has two pivots, it has two kinematic models, one for each pivot. The derivation of the kinematic model for pivot A can be presented as follows. The velocity of the millirobot's center in the millirobot frame v_C^R is defined using the relative velocity between points C and A such as:

$$v_C^R = v_A^R + \bar{\omega} \times r_{C/A}^R \tag{1.48}$$

where v_A is equal to zero since the point A is fixed pivot, $\bar{\omega}$ is the millirobot angular velocity about z-axis, and $r_{C/A}^R$ is the position vector between point C and A. Then, $\bar{\omega}$ and $r_{C/A}^R$ are defined as follows:

$$\bar{\omega} = \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \tag{1.49}$$

$$r_{C/A}^R = \begin{bmatrix} 0 \\ -\frac{L}{2} \\ 0 \end{bmatrix} \tag{1.50}$$

where ω is the angular velocity about the $z - axis$ and L is the millirobot length. So, v_C^R

simplifies to

$$v_C^R = \begin{bmatrix} \frac{L\omega}{2} \\ 0 \\ 0 \end{bmatrix} \tag{1.51}$$

Now, using Eq. (1.47), the millirobot velocity in the inertial frame v_C^I can be obtained as follows:

$$v_C^R = \begin{bmatrix} \frac{L}{2}\omega\cos(\theta) \\ \frac{L}{2}\omega\sin(\theta) \\ 0 \end{bmatrix} \tag{1.52}$$

Similarly, for pivot B, Eqs. (1.48)-(1.52) are used to obtain the millirobot velocity in the inertial frame v_C^I. Where the relative velocity between the points C and B is defined as follows

$$v_C^R = v_B^R + \bar{\omega} \times r_{C/B}^R \tag{1.53}$$

where v_B is equal to zero since the point B is fixed pivot and $r_{C/B}$ is the position vector between the points C and B. Then, $r_{C/B}$ is defined as follows:

$$r_{C/B}^R = \begin{bmatrix} 0 \\ \frac{L}{2} \\ 0 \end{bmatrix} \tag{1.54}$$

Therefore, the millirobot velocity in the inertial frame v_C^I is described as follows:

$$v_C^R = \begin{bmatrix} -\frac{L}{2}\omega\cos(\theta) \\ -\frac{L}{2}\omega\sin(\theta) \\ 0 \end{bmatrix} \tag{1.55}$$

The pivot walker kinematic model can be summarized as follows:

$$\dot{q}_I = \begin{cases} \left.\begin{array}{c} \dot{x} = \frac{L}{2}\omega\cos(\theta) \\ \dot{y} = \frac{L}{2}\omega\sin(\theta) \\ \dot{\theta} = \omega \end{array}\right\} & \text{for pivot A} \\ \\ \left.\begin{array}{c} \dot{x} = -\frac{L}{2}\omega\cos(\theta) \\ \dot{y} = -\frac{L}{2}\omega\sin(\theta) \\ \dot{\theta} = \omega \end{array}\right\} & \text{for pivot B} \end{cases} \tag{1.56}$$

where $\dot{x}$ and $\dot{y}$ are the millirobot linear velocities, $\dot{\theta}$ is the millirobot angular velocity. As can be seen, the kinematic model is a hybrid model that contains continuous-time functions and discrete events (e.g the switching between the two models). Hybrid systems [47] are harder to control. As we have control over the switching between two pivots, the switching can be modeled as input to the kinematic model that results in combining the two kinematic models into a single model as presented:

$$\dot{q}_I = \begin{bmatrix} \frac{L}{2}\sigma\omega\cos(\theta) \\ \frac{L}{2}\sigma\omega\sin(\theta) \\ \omega \end{bmatrix} \tag{1.57}$$

where σ is the switching control input between the models, If $\sigma = 1$, the pivot A is active, whereas $\sigma = -1$, means that B is the active pivot. Value of the parameter σ is constrained by either 1 or -1 depending on the desired active pivot.

Finally, the positions of pivots A and B are obtained using vector projection, where pivot A position is calculated as follows:

$$x_A = x - \frac{L}{2}\sin\theta \tag{1.58}$$

$$y_A = y + \frac{L}{2}\cos\theta \tag{1.59}$$

similarly, the position of pivot B is

$$x_B = x + \frac{L}{2}\sin\theta \tag{1.60}$$

$$y_B = y - \frac{L}{2}\cos\theta \tag{1.61}$$

1.14 Locomotion Analysis

The locomotion of the millirobot is composed of successive steps. Each step can be described as a motion about a pivot point. The location of the pivot point alternates between two contact points with the ground surface. Figure 1.19 schematically shows a two-step progression of this type of locomotion. During the first step, pivot A is active and no motion occurs there. Meanwhile, pivot B is free to move with the millirobot body. Once the main link rotates into the desired angle the active pivot is switched to pivot B and now pivot A is free to move with the main link. This process is repeated to generate forward locomotion.

In general, walking consists of some successive steps. To analyze the locomotion of the millirobot, we define a set of gait parameters. The sweep angle θ_s is defined as the angle between two successive steps. The number of switches between the two pivots is defined as the number of steps N_s. The sweep angle has a significant impact on the motion, the number of steps, and the travel distance. In order to demonstrate the effect of the sweep angle, a locomotion algorithm is designed to control the millirobot to walk along a straight line. This locomotion algorithm was designed such that the millirobot starts from initial position, then takes a step with half of the desired sweep angle. Subsequently, the pivot is switched and followed by a step with a full desired sweep angle. Finally, the pivot is switched again to perform the next step on the other pivot. The straight-line locomotion is shown in Algorithm 1.20.

Figure 1.21 shows different robot trajectories for different sweep angles in the range of 1° to 360°. As can be seen from Fig. 1.21, the sweep angle has a significant effect on the number of steps. Whereas, the number of steps is very large for small angles and gradually reduces for increasing values of the sweep angle. Another observation is that the millirobot path is almost a straight line at smaller angles, while curved paths appear at larger sweep

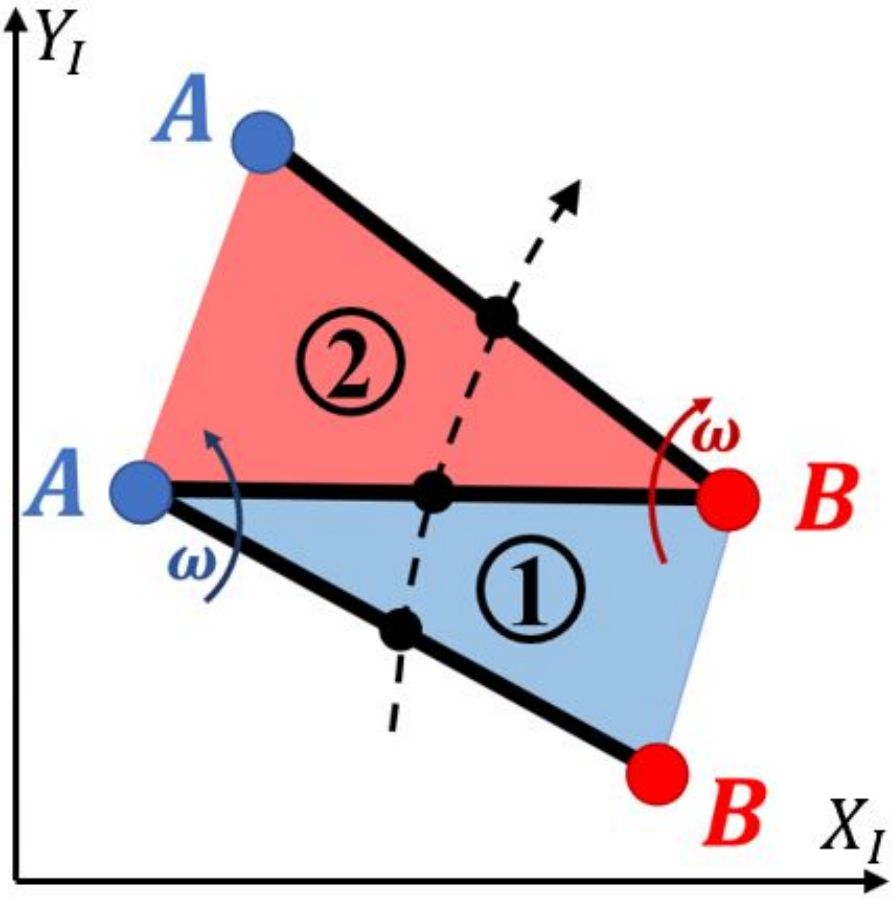


Figure 1.19: A two-step angular progression in pivot walking.

angles. A full parametric analysis is conducted to see the effect of the sweep angle on the number of steps and travel distance.

Figure 1.22 shows the number of steps as a function of the sweep angle for a travel distance of 6 cm. At lower sweep angles, the number of steps is very large with a singularity at $\theta_s = 0$. Increasing the sweep angle results in a fewer number of steps, As can be seen in Fig. 1.21, after about 200°, the number of steps starts increasing as the contribution of the steps to the forward motion diminish. At $\theta_s = 360°$, there is another singularity since the millirobot rotates around its own central axis. Also, a higher number of steps results in longer completion times. Figure 1.23 depicts the millirobot's travel distance as a function of the sweep angle. Ideally, the travel distance is equal to the path length at lower sweep angles. But increasing the sweep angle results in drifts from the path length leading to higher travel

Algorithm 1: Straight-line locomotion

$N_s = 0; \quad \theta_0 = 0;$
$\omega = \omega_0; \quad \sigma = 1 \ ;$
while $y_d - y > 0$ **do**
  if $N_s = 1$ **then**
    $\theta_s = \frac{\theta_d}{2};$
  else
    $\theta_s = \theta_d;$
  end
  if $|\theta - \theta_0| = \theta_s$ **then**
    $\sigma = -\sigma;$
    $\omega = -\omega;$
    $\theta_0 = \theta;$
  end
  q = robotKinemaitc (q, σ, ω);
  $N_s = N_s + 1;$
end

Figure 1.20: Straight-line locomotion.

distances, and therefore, to longer task completion times.

These two parametric analysis results can be used in designing and optimizing control algorithms for the pivot walker robot. For example, a small sweep angle can be used to perform accurate tasks, but small sweep angles result in a higher number of steps. Therefore, the process becomes a trade-off between accuracy and task completion time.

1.15 Control Design

Based on the kinematic model presented in Eq. (1.57), the millirobot motion can be governed using two control inputs σ and ω. The input σ determines the active pivot that the millirobot rotates about and ω defines how fast the millirobot will rotate about the active pivot. In this section, we present a control algorithm for a pivot walking robot to track an arbitrary trajectory. The goal of this controller is to minimize the error e between the millirobot center (x, y) and the desired trajectory (x_d, y_d). Figure 1.24 shows the millirobot

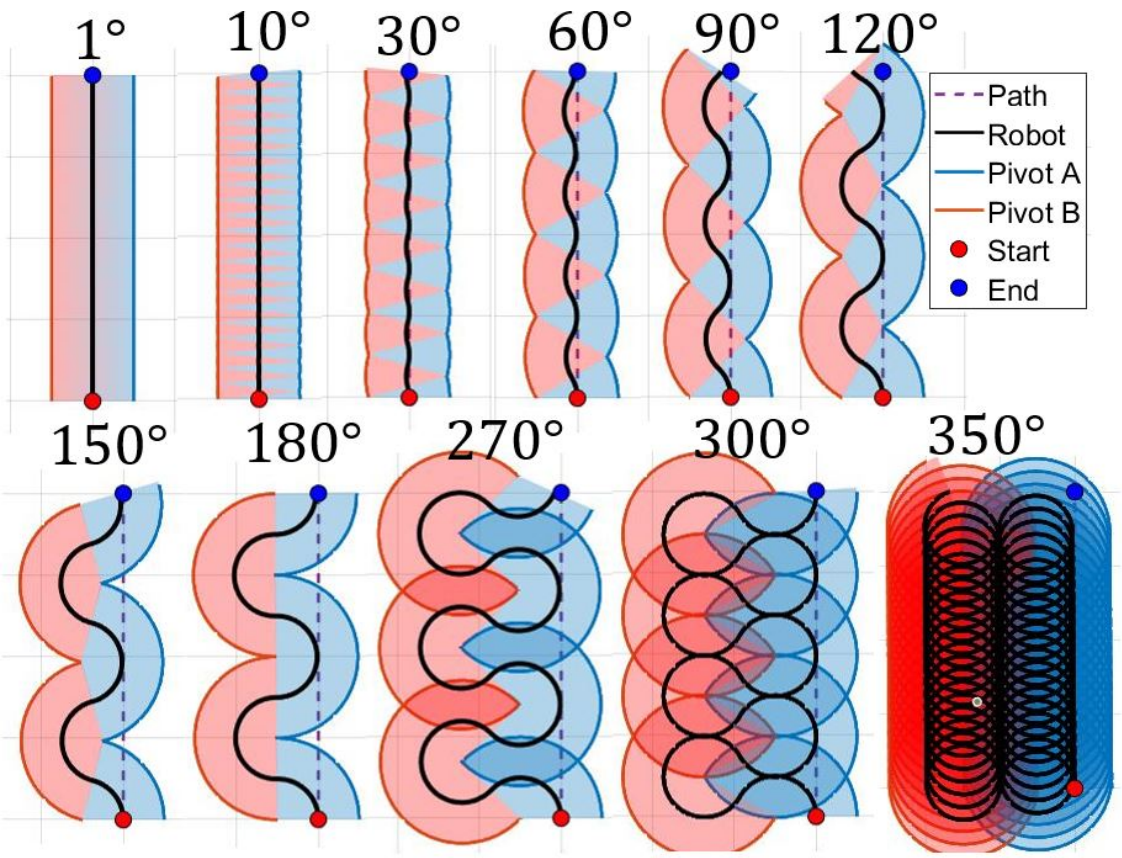


Figure 1.21: Different robot paths for different sweep angles. The robot length is 10 mm and the desired path length is 40 mm. The purple dashed line represents the straight path, the center of the millirobot trajectory is shown as a thick black line. The blue and red lines represent the pivots A and B respectively. Blue and red areas indicate that the millirobot rotates about pivots A and B respectively. The change in color in these areas represents the number of switches between the two pivots, hence the number of steps.

and the desired trajectory. The challenge in designing a control algorithm for such systems is how to handle the switching between the two kinematic models. There are two ways to solve this problem. First, pre-plan the motion using algorithms that take into consideration the millirobot kinematics (e.g humanoid and bipedal robots). Second, design an algorithm that can solve the planning and control problem, simultaneously. Here, we will focus on the second approach. Two control algorithms are proposed to track the desired trajectory without pre-planning the motion.

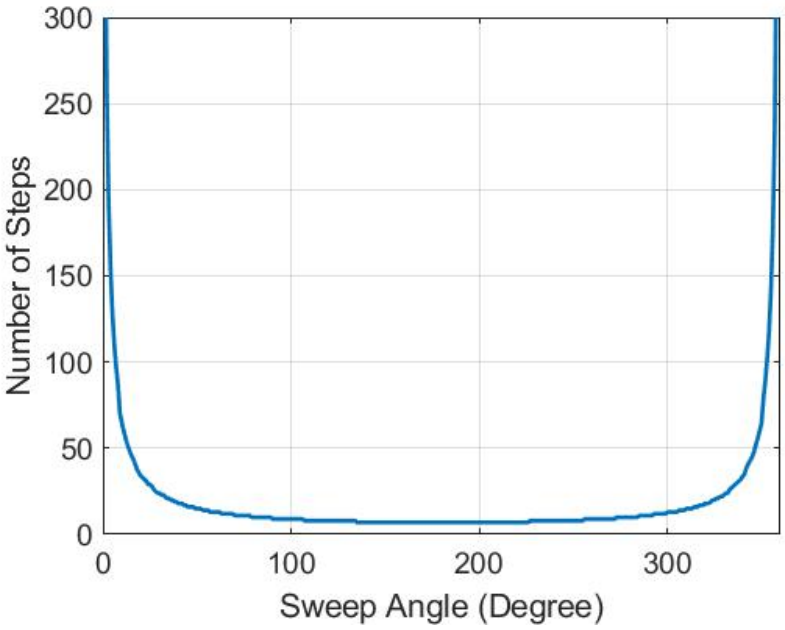


Figure 1.22: Variation of number of steps to travel 6 cm as a function of sweep angle.

1.15.1 Geometric based controller

As discussed earlier, the challenge in controlling a walking robot is how to design the control algorithm that switches between the pivots to minimize the error. We propose to use the distance between two pivots and the desired path, and based on the distance, the millirobot uses the nearest pivot as the active pivot and rotates about it. The idea behind this algorithm is that the millirobot rotates about the nearest pivot (e.g. A) to the trajectory until the other pivot (e.g. B) becomes the nearest one. Then, the pivot is switched to B until A becomes the nearest pivot to the trajectory, and so on. Expressions for the distances between the pivots and the desired trajectory are shown in Fig. 1.24, and can be obtained as follows:

$$d_{Ad} = \sqrt{(x_A - x_d)^2 + (y_A - y_d)^2)} \tag{1.62}$$

$$d_{Bd} = \sqrt{(x_B - x_d)^2 + (y_B - y_d)^2)} \tag{1.63}$$

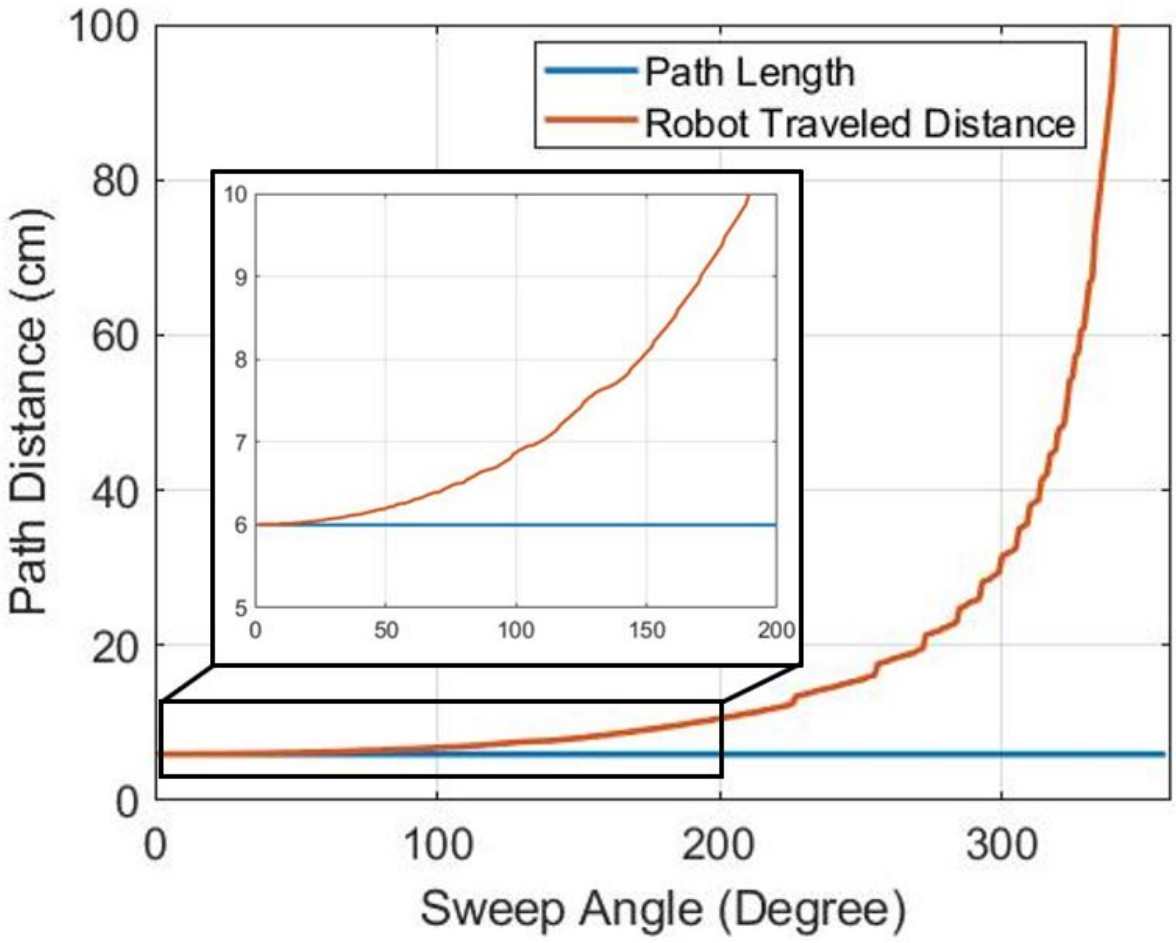


Figure 1.23: Variation of robot traveled distance as function of sweep angle.

Then, the active pivot is:

$$\sigma = \text{sign}(d_{Bd} - d_{Ad}) \tag{1.64}$$

where $\sigma = 1$ if d_{Bd} is greater than d_{Ad} and $\sigma = -1$ if d_{Bd} is less than d_{Ad}.

After selecting the active pivot that minimizes the error, a proportional controller for the angular velocity ω is formulated as follows:

$$\omega = k\,\sigma\,e_n \tag{1.65}$$

where k is a positive gain, and e_n is the norm of error between the millirobot center and the

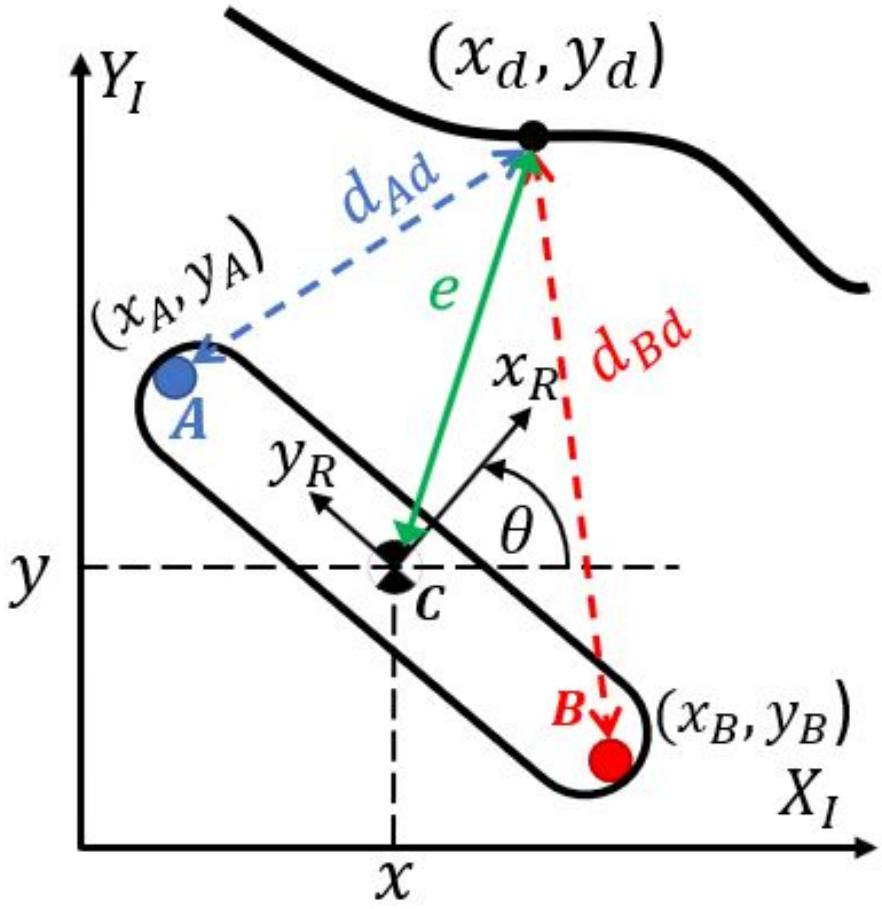


Figure 1.24: Control problem description.

desired trajectory given by:

$$e_n = \sqrt{(x - x_d)^2 + (y - y_d)^2)} \tag{1.66}$$

The sign of σ plays an important role in this control law (Eq. (1.65)) to specify in what direction the millirobot should rotate. Finally, this control algorithm integrated with the sweep angle concept and shown in Algorithm 1.25, where the switching between the two pivots is constrained by completing the sweep angle.

Algorithm 2: Geometry based controller

Initialization ;
$q = q0$;
$e_n = \sqrt{(x - x_d)^2 + (y - y_d)^2)}$;
$d_{Ad} = \sqrt{(x_A - x_d)^2 + (y_A - y_d)^2)}$;
$d_{Bd} = \sqrt{(x_B - x_d)^2 + (y_B - y_d)^2)}$;
$\sigma = \text{sign}(d_{Bd} - d_{Ad})$;
$\omega = \sigma\, e_n$;
while $t < t_f$ **do**
  if $\theta - \theta_0 = \theta_s$ **then**
    $d_{Ad} = \sqrt{(x_A - x_d)^2 + (y_A - y_d)^2)}$;
    $d_{Bd} = \sqrt{(x_B - x_d)^2 + (y_B - y_d)^2)}$;
    $\sigma = \text{sign}(d_{Bd} - d_{Ad})$;
  end
  $e_n = \sqrt{(x - x_d)^2 + (y - y_d)^2)}$;
  $\omega = k\, \sigma\, e_n$;
  q = robotKinemaitc (q, σ, ω);
  $t = t + \Delta t$;
end

Figure 1.25: Geometric based algorithm.

1.15.2 Optimization based controller

In controls theory, the control law's purpose is to minimize the error between the millirobot and the desired states. In other words, the control problem is an optimization problem with the following cost function J.

$$J = \frac{1}{2}(q - q_d)^2 \tag{1.67}$$

and in more details,

$$J = \frac{1}{2}((x - x_d)^2 + (y - y_d)^2 + (\theta - \theta_d)^2)) \tag{1.68}$$

Our goal here is to minimize the cost function, by optimizing the control inputs σ and ω. Using the kinematic model (Eq. (1.57)), the sensitivity of the cost function concerning control inputs is obtained using the chain rule. This control algorithm has a similar structure to a neural network, but it is based on the kinematic network. This network consists of several layers; the first layer is the input nodes and the last layer is the millirobot position and orientation. The intermediate layers represent the kinematic connection between the input and the output. Finally, the back-propagation depends on the same concept of the sensitivity of the objective function for control inputs.

The derivation of the proposed control algorithm is presented next. The sensitivity of the cost function with respect to the angular velocity $\frac{\partial J}{\partial \omega}$ is derived as:

$$\frac{\partial J}{\partial \omega} = \frac{\partial J}{\partial x}\frac{\partial x}{\partial \omega} + \frac{\partial J}{\partial x}\frac{\partial x}{\partial \theta}\frac{\partial \theta}{\partial \omega} + \frac{\partial J}{\partial y}\frac{\partial y}{\partial \omega} + \frac{\partial J}{\partial y}\frac{\partial y}{\partial \theta}\frac{\partial \theta}{\partial \omega} + \frac{\partial J}{\partial \theta}\frac{\partial \theta}{\partial \omega} \tag{1.69}$$

where the partial derivatives are obtained as follows

$$\begin{aligned}
&\frac{\partial J}{\partial x} = x - x_d && , \frac{\partial J}{\partial y} = y - y_d && , \frac{\partial J}{\partial \theta} = \theta - \theta_d \\
&\frac{\partial x}{\partial \omega} = \Delta t\, \sigma \cos\theta && , \frac{\partial y}{\partial \omega} = \Delta t\, \sigma \sin\theta && , \frac{\partial \theta}{\partial \omega} = \Delta t \\
&\frac{\partial x}{\partial \theta} = -\Delta t\, \sigma\omega \sin\theta && , \frac{\partial y}{\partial \theta} = \Delta t\, \sigma\omega \cos\theta &&
\end{aligned} \tag{1.70}$$

Next, the sensitivity of the cost function with respect to the active pivot $\frac{\partial J}{\partial \sigma}$ becomes:

$$\frac{\partial J}{\partial \sigma} = \frac{\partial J}{\partial x}\frac{\partial x}{\partial \sigma} + \frac{\partial J}{\partial y}\frac{\partial y}{\partial \sigma} + \frac{\partial J}{\partial \theta}\frac{\partial \theta}{\partial \sigma} \tag{1.71}$$

where the partial derivatives are obtained as follows

$$\frac{\partial x}{\partial \sigma} = \Delta t\, \omega \cos\theta \qquad , \frac{\partial y}{\partial \sigma} = \Delta t\, \omega \sin\theta \qquad , \frac{\partial \theta}{\partial \sigma} = 0 \tag{1.72}$$

Using the gradient descent, the control laws are

$$\omega_0 = \omega_0 + \eta \frac{\partial J}{\partial \omega} \tag{1.73}$$

$$\sigma_0 = \sigma_0 + \eta \frac{\partial J}{\partial \sigma} \tag{1.74}$$

where η is the learning rate. Then, saturation and sign functions are used to constrain the control inputs.

$$\omega = \text{sat}(\omega_0) \tag{1.75}$$

$$\sigma = \text{sign}(\sigma_0) \tag{1.76}$$

This control algorithm learns and optimizes the control inputs to minimize the cost function, which is the error in this case. Also, this algorithm has an advantage over the previous one because it is able to regulate the orientation of the millirobot.

1.16 Simulation Results

We test the performance of the proposed controllers by conducting a set of simulation runs in different cases. In each case, we use an eight-shape path as the desired trajectory. The initial condition of the millirobot used in the simulation runs is $q_I = [0\text{cm} \ -4.2\text{cm} \ \frac{\pi}{2}]^T$, and the effect of varying sweep angle is studied. The 10° and 30° sweep angles are considered for the first and second cases respectively. In both cases, the following eight-shape trajectory is used.

$$x_d = -4 \sin(0.1\, t) \tag{1.77}$$

$$y_d = -4 \cos(0.05\, t) \tag{1.78}$$

It is worth to note that the controllers are tested in the absence of tilt angle and number of steps constraints. Also, the step time is fixed at 0.1 second.

Figure 1.26(a) depicts the simulation results for both cases. As it can be seen, the control methods with smaller θ perfectly track the desired trajectory. Figure 1.26(b) shows the norm of the tracking error e_n, which is presented in Eq. (1.66). Therefore, both controllers perform

well, but the optimization based controller shows better performance. Also, in terms of the number of steps, the geometric and optimization based controllers are taking 296 and 269 steps to track the desired trajectory, respectively.

In order to test the robustness of the proposed controllers, trajectory tracking is tested with a 30° sweep angle. As one can see in Fig. 1.26(a), two control methods track the desired trajectory with more deflections. The norm of the tracking error e_n illustrated in Fig. 1.26(b), shows these higher errors. Although two controllers perform well, the optimization based controller outperforms the geometric based one again. Also, in terms of the number of steps, two controllers have approximately a similar number of steps; 75 steps are taken in the optimization method and 71 steps in the geometric one.

Clearly, the sweep angle affects the controllers' performance because it constrains the millirobot kinematics. As we expected, a comparison of both cases shows that larger sweep angles result in a fewer number of steps but the tracking error performance deteriorates.

1.17 Experimental Results

The experimental results section will be divided into two parts: the first will discuss the experimental setup and signal processing; the second part will discuss the tracking performance of the proposed controllers, and the experimentally parametric analysis.

1.17.1 Experimental setup and signal processing

Figure 1.27 shows an overview of the experimental setup. In the experiments, we use a 10-mm in length millirobot, which is 3D printed using the photo-polymer resin (see Fig. 1.4(c)). The nested electromagnetic Helmholtz coil actuates this millirobot.

The pivot walker position and orientation are obtained via visual feedback. Image processing techniques are utilized to estimate the pivot walker pose from the video stream. A snapshot of video steam was taken at every time step. Then, a hue saturation value (HSV) filter was applied to detect the pivot walker color. Subsequently, we use a region filter to detect a rectangle that defines the millirobot based on its area and centroid. Finally,a filter is used to combine the predicted states (x_p, y_p, and θ_p) from kinematic model with the measured states (x_m, y_m, and θ_m) from the camera. this filter is helpful to reduce noise and avoid in position and orientation feedback. The overall system structure can be summarized

in Fig 1.28.

1.17.2 Tracking performance and parametric analysis

In order to validate our proposed control schemes, we conduct a set of different experiments and analyze the data to study the locomotion dynamics. Three different parameters are played the main role in the motion outcomes: tilt angle(α), sweep angle (θ), and the generating desired path time step (Δt). We perform parametric analysis on these factors and study the effect of them on the error between the experimental results and the desired path. Here, we define a variable S as a set of the values of three parameters $S = [\theta, \alpha, \Delta t]$; for example if, in an experiment, we use sweep angle as 30°, tilt angle as 15°, and time step as 0.1, the set will be $S = [30, 15, 0.1]$. Note that the developed kinematic model does not take into account the effect of the tilt angle (α) so it is also useful to study the gradual change of this parameter.

To conduct the experimentally parametric analysis on our proposed controllers, we vary the sweep angle between 10 to 40 degrees and the tilt angle between 20 to 30 by increment 5 and choose different values for $\Delta t = \{0.1, 0.2, 0.3\}$. In total, we have in total 63 different experimental sets. We consecutively conduct ten experiments for each set. First, for each set and experiment, we calculate the means of the errors between the center of millirobot and the desired trajectory over the entire trajectory. Then, we normalize these means by the length of the millirobot and compute the mean for each set over all ten experiments denoting by ME.

For the first step, we choose the geometric based controller to analyze the effect of sweep and tilt angles and the time step on ME. We conduct ten experiments for each experimental set and obtain the mean values. Figure 1.29 shows the variation of ME for different tilt angles and step time when the sweep angle is fixed at ten degrees. As one can see in Fig. 1.29, the error follows a consistent trend by varying the tilt angle. First, the error is decreasing to reach its minimum value of around $\alpha = 24.5°$, then it increases. This trend is completely related to the stickiness and the slippage of the pivot points on the surface.

In Fig. 1.29(a), the dashed line in the middle at $\alpha = 24.5°$ represents the point that the minimum value of the ME occurs. On the left side of this line, the error increases due to

more sticky motions on the pivot points. On the other hand, more slippage on the pivot points due to a higher tilt angle causes more error on the right side of the line. Also, for each time step, the slippage problem brings out more error than the stickiness on the pivot point. This pattern can also be seen in Fig. 1.29(b). In each time step, the minimum takes place near $\alpha = 25°$, and tilt angles more than 25 degrees have more error than the angles less than 25 degrees.

Figure 1.30 depicts the actual trajectory tracking of the millirobots, which represents the results of Fig. 1.29. In this figure, one can recognize the effects of the stickiness and slippage of the pivot points and varying the time step. As one can see, by increasing the tilt angle, more slippage occurs. By increasing the step time, generating the desired path becomes faster than the reaction of the millirobot, and it can not follow the path. This lag is completely obvious if someone pays attention to the final position of the millirobot. At $\Delta t = 0.3$s, the distance between the final and initial positions is greater than the others for every tilt angle.

Figure 1.31 depicts the parametric analysis over different sweep angles. The minimum MEs are near 25 degrees for tilt angle in most of the cases. The surface of the experimental set-up workspace does not have a uniform coefficient of fiction. Thus, we see some inconsistencies in the experimental errors. But, the pattern of increasing in ME happens when the tilt angle is varied up and down. In Fig. 1.31, at $\alpha = 30°$, by increasing the sweep angle, the ME is decreasing. This happens because when the millirobot rotates more in each time step, it can overcome more on the stickiness problem, without adding the slippage problem; but by increasing the time step, the slippage issue appears and as shown in Fig. 1.31(a) to (c), the overall mean error increases. The optimum value for sweep angle at $\alpha = 30°$ is approximately 35° and at $\alpha = 20°$ is 30°. We can conclude that the optimum sweep angle can be $30° \leq \theta_0 \leq 35°$.

Figure. 1.32 shows the effect of the sweep angle in the trajectory tracking represented in Fig. 1.31. One can find the video of these experiments as supplementary materials.

To compare both controllers and analyze the error distribution, we choose three different sets of main parameters and conduct 100 experiments for each set and controller. The sets

are:

$$S_1 = \{25, 25, 0.1\}$$
$$S_2 = \{30, 25, 0.1\} \tag{1.79}$$
$$S_3 = \{30, 30, 0.1\}$$

We should note that the time step is chosen to be constant in order to check the differentiation of stickiness and slippage factors. Figure 1.33 shows the histogram distribution of normalized error over 100 experiments. As we expect, it shows a semi-bell-shaped curve and more Weibull distribution. The peak of the curve corresponds to the ME. As one can see, by increasing the tilt angle, the distribution shifts to left and shows more error due to more slippage, and also shows a wider distribution, smaller standard deviation, and smaller peak. Also, a higher sweep angle causes less mean value. Table 1.1 presents the variance and standard deviation (SD) of each set. In the overall view, as we showed in Fig. 1.26(b), the optimal controller has less error than the geometric based controller. It can be concluded via less mean values of histogram for the optimal control scheme. Figure. 1.32 shows typical examples of these experiments. The out-performance of the optimization based controller can be also recognizable.

Table 1.1: The variance and standard deviation of experimental results of different sets when the time step is fixed at 0.1 second.

Controller	θ (deg)	α (deg)	Var.	SD
Gb	25	25	41.8677	6.47053
Ob	25	25	47.9418	6.47625
Gb	30	25	98.4696	9.92318
Ob	30	25	99.291	9.96449
Gb	30	30	14.0265	3.46792
Ob	3 0	30	14.6243	3.82418

1.18 Conclusion

In this section, kinematic modeling and control for pivot walking millirobots were addressed. We proposed two control algorithms to guide the pivot walker while it follows the desired path. The first controller was a proportional-geometric based approach. In this approach, the controller activated the nearest pivot point to the desired trajectory. Then, it used a proportional controller to regulate the angular velocity about that pivot point. The second method was based on optimization approach. The gradient descent algorithm was used to optimize the active pivot and the angular velocity and minimize the tracking error between the millirobot and the desired trajectory.

First, we performed a numerical parametric analysis to investigate the effect of the sweep angle on the number of steps and travel distance. We checked the performance of the tracking algorithms within two simulation scenarios, one with 10 degrees sweep angle, the other one with 30 degrees. In both scenarios, two algorithms were tested without placing constraints on the tilt angle and step time. We showed that the optimization-based controller outperformed the proportional-geometric approach. It had a faster convergence time, smaller tracking error, but a higher number of steps. In addition, we conducted many experimental tests to verify our proposed controllers. We analyzed the effect of sweep angle, tilt angle, and step time in trajectory tracking. We showed that at a certain tilt angle, we have a minimum tracking error. Although in the smaller angle, the stickiness of the pivot point on the ground cause more error, in higher angles, more tracking error is a result of the slippage at the pivot points.

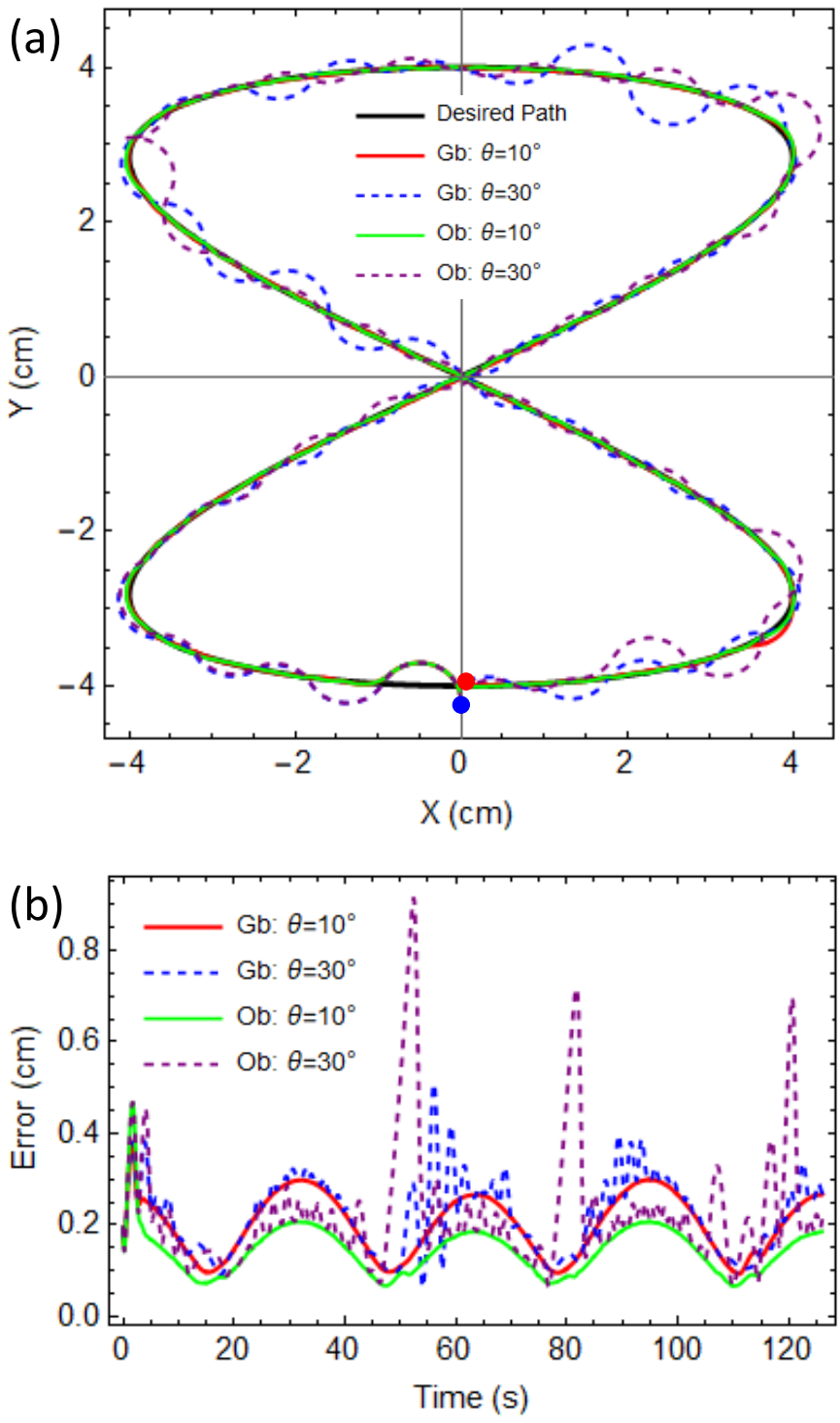


Figure 1.26: Simulation results of tracking an eight-shape path with different sweep angles. **(a)** Tracking the desired path with sweep angles 10 and 30 degrees. The Blue and red circles show the start and end positions, respectively. The black, red, dashed blue, green, and dashed purple are the desired trajectory, the simulation results of the geometric based controller of 10 and 30 degrees and the simulation results of optimization based controller of 10 and 30 degrees, respectively. **(b)** Time histories of the error between the center of the robot and the desired path. In the legend, Gb and Ob denote geometric and optimization based controllers respectively.

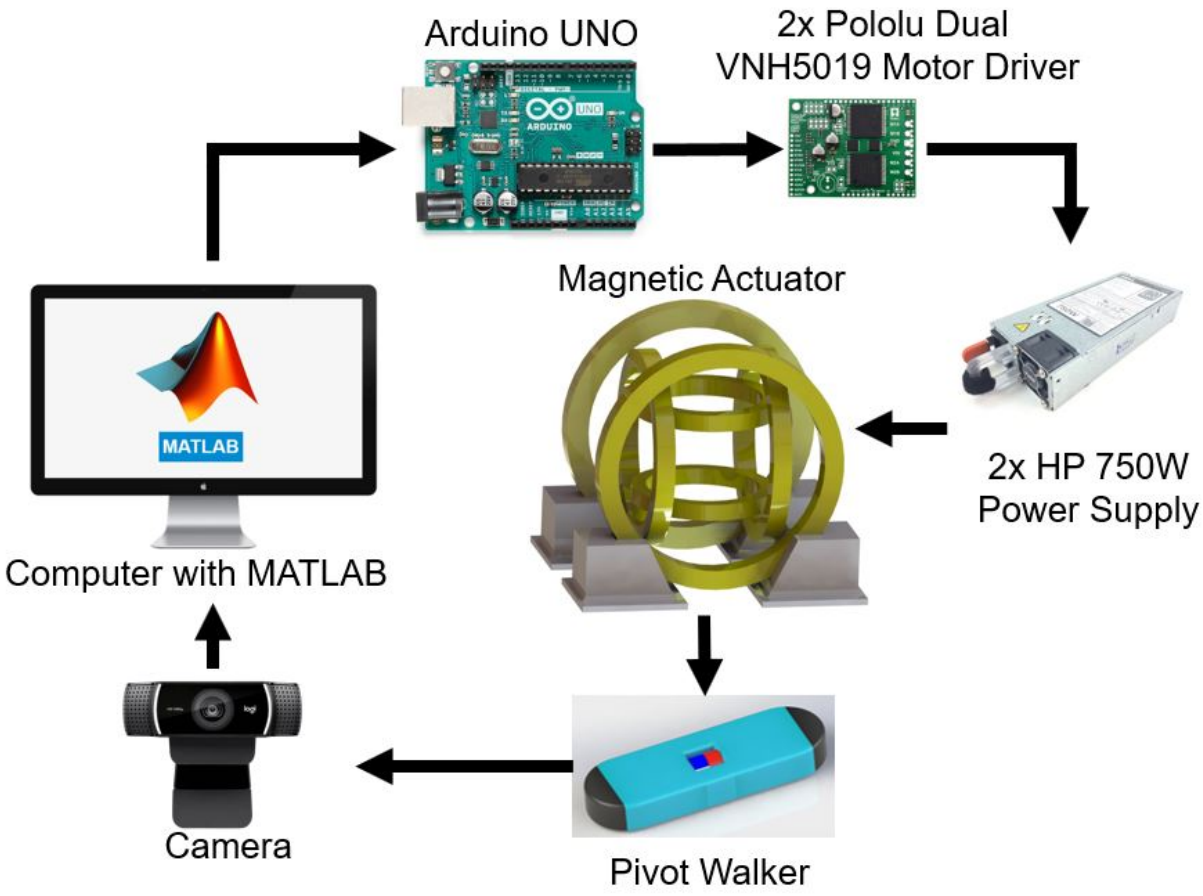


Figure 1.27: Overview of the experimental setup.

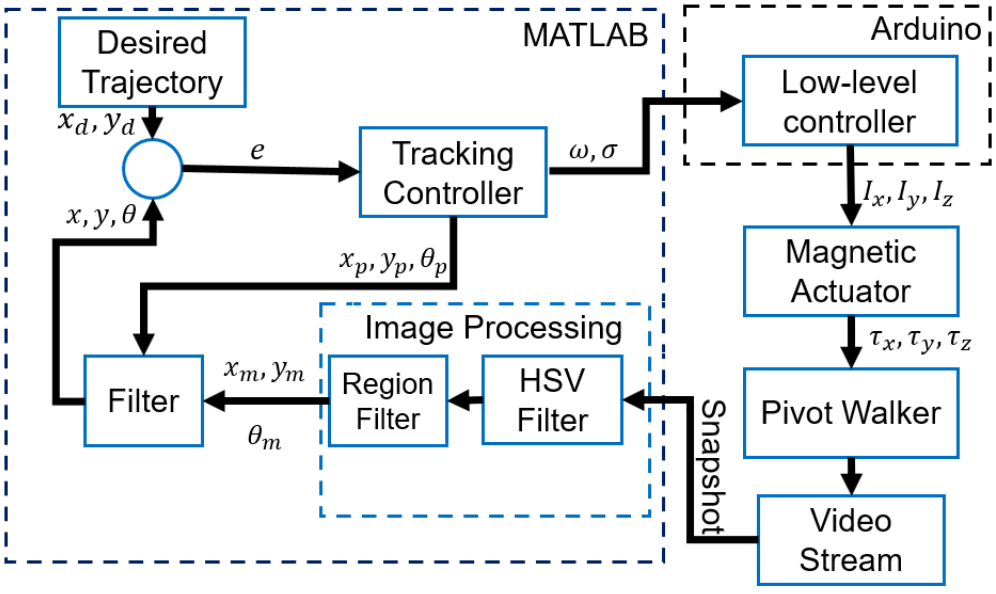


Figure 1.28: The system block diagram.

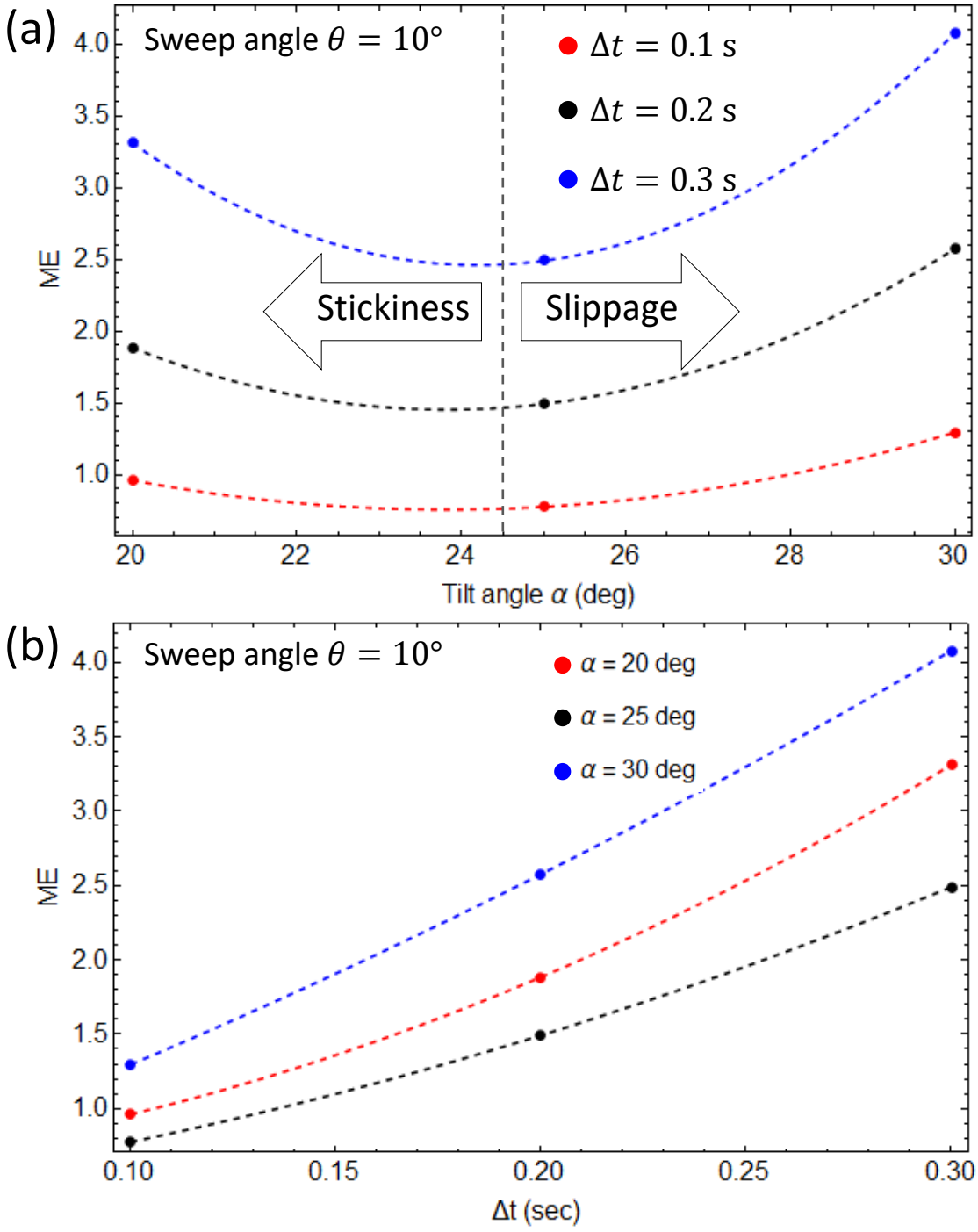


Figure 1.29: Experimentally parametric analysis of ME when the sweep angle $\theta = 10°$ is constant. **(a)** Variation of ME over different tilt angle. **(b)** Variation of ME over different step time.

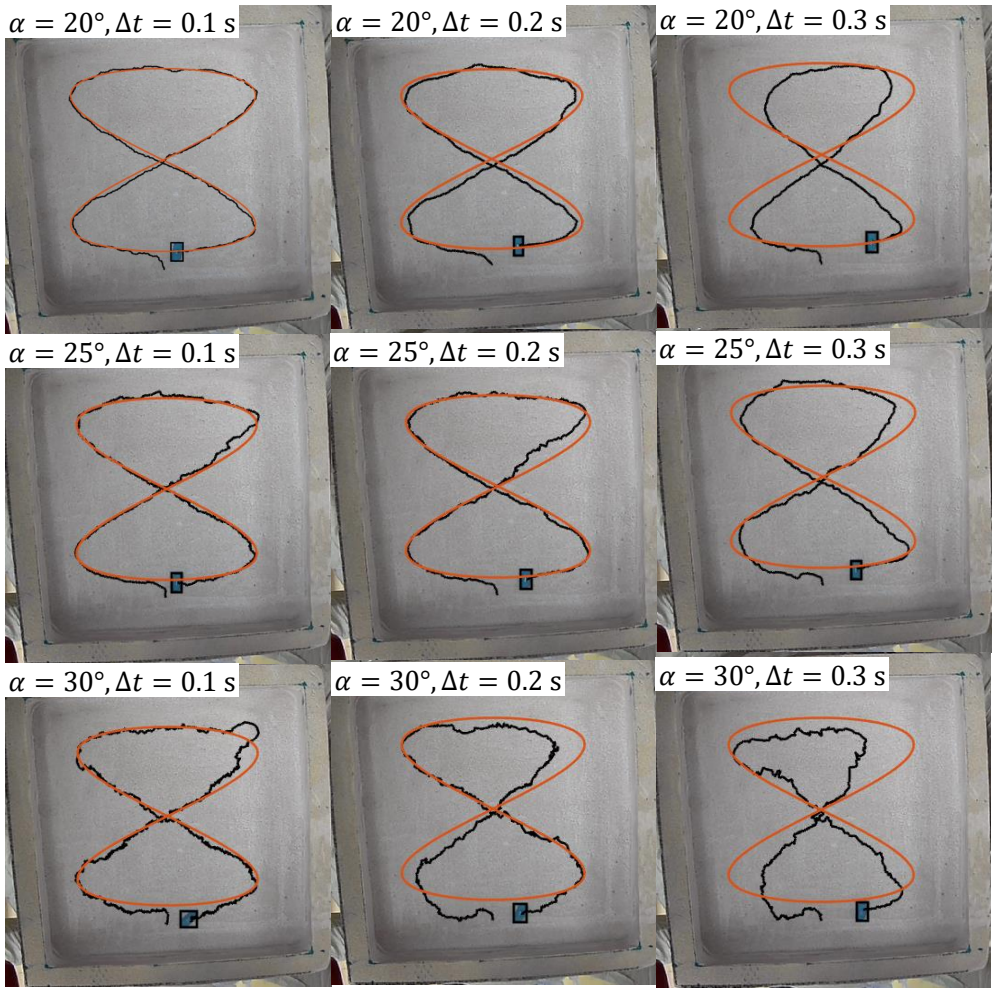


Figure 1.30: Trajectory tracking experimental results when the sweep angle is fixed at $\theta = 10°$. The orange and black lines are desired trajectory and the actual trajectory of the center of the millirobot respectively. The final position of the millirobot is indicated by a back rectangle at the end of the trajectory.

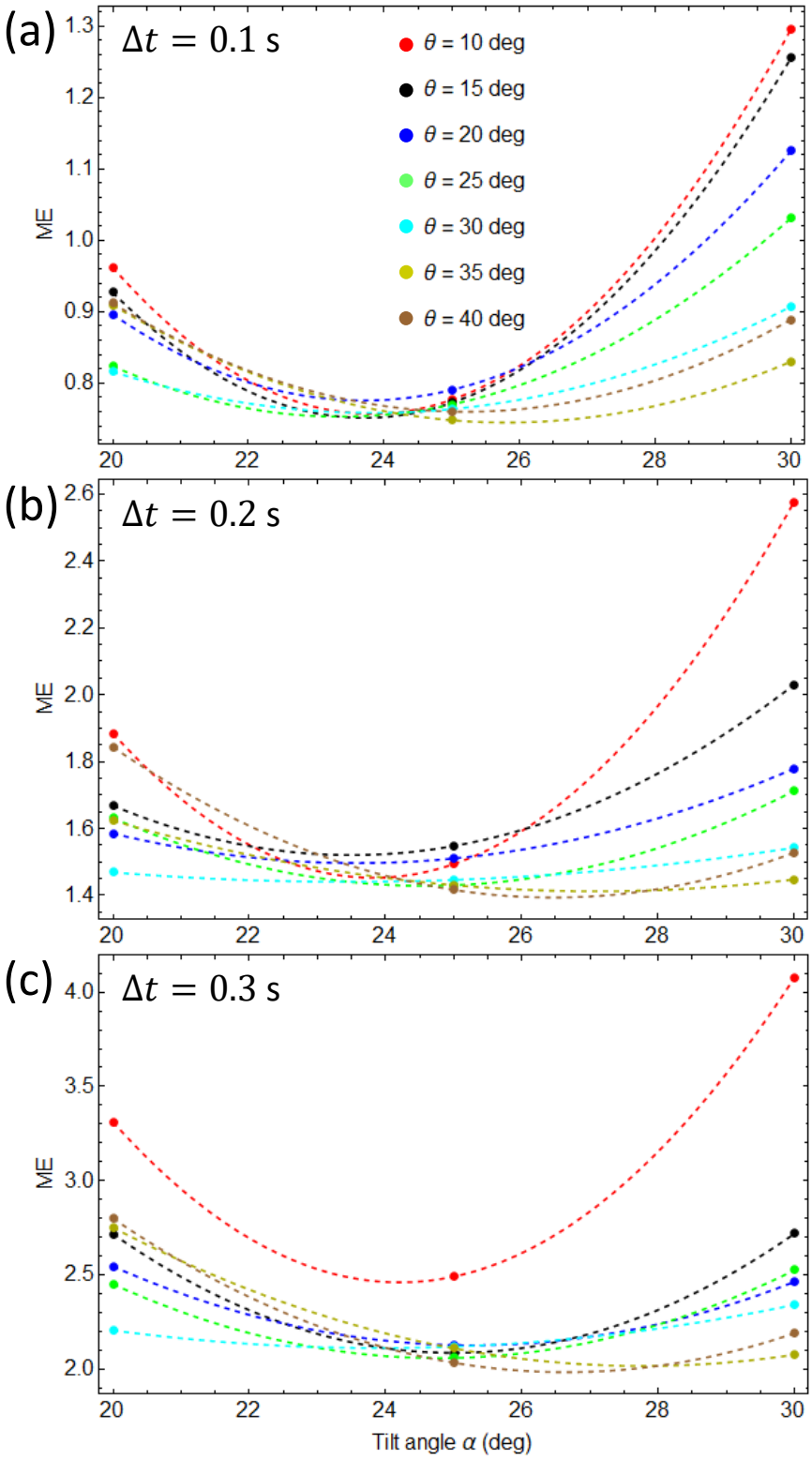


Figure 1.31: Experimentally parametric analysis of ME when the tilt angle is varying for different sweep angles.**(a)** Variation of ME for $\Delta t = 0.1$ s.**(b)** Variation of ME for $\Delta t = 0.2$ s.**(c)** Variation of ME for $\Delta t = 0.3$ s.

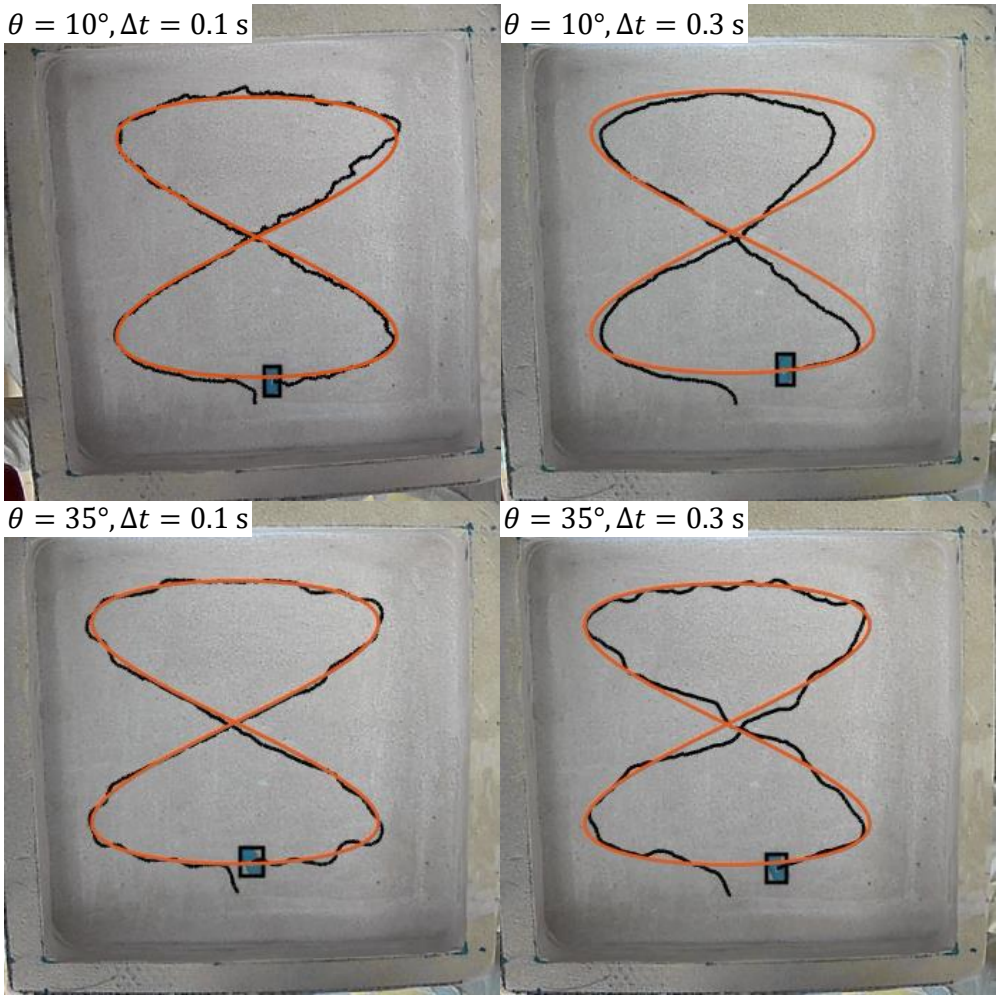


Figure 1.32: Trajectory tracking experimental results for different sweep angles without the tilt angle constraint ($\alpha = 25°$). The orange and black lines are desired trajectory and the actual trajectory of the center of the millirobot respectively.

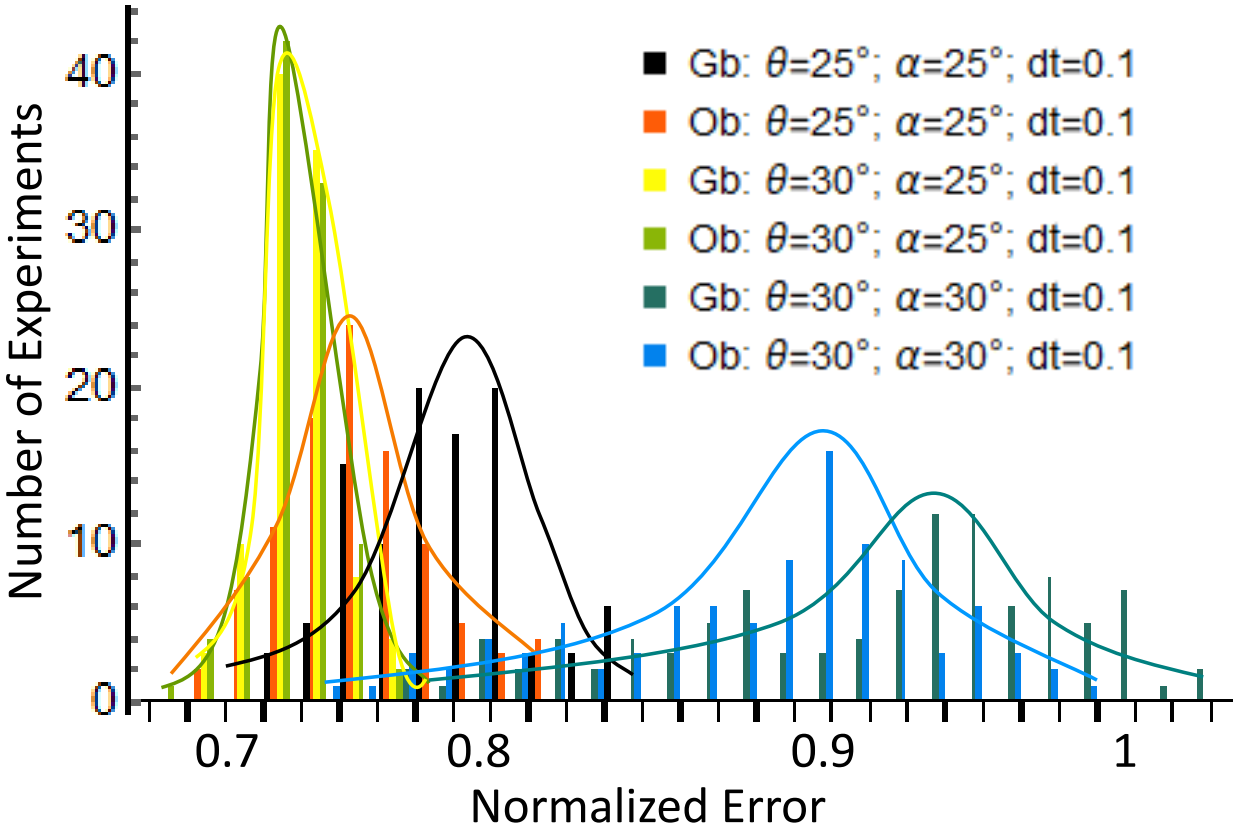


Figure 1.33: The Experimentally error distribution for different controllers. In the legend, Gb and Ob denote geometric and optimization based controllers respectively.

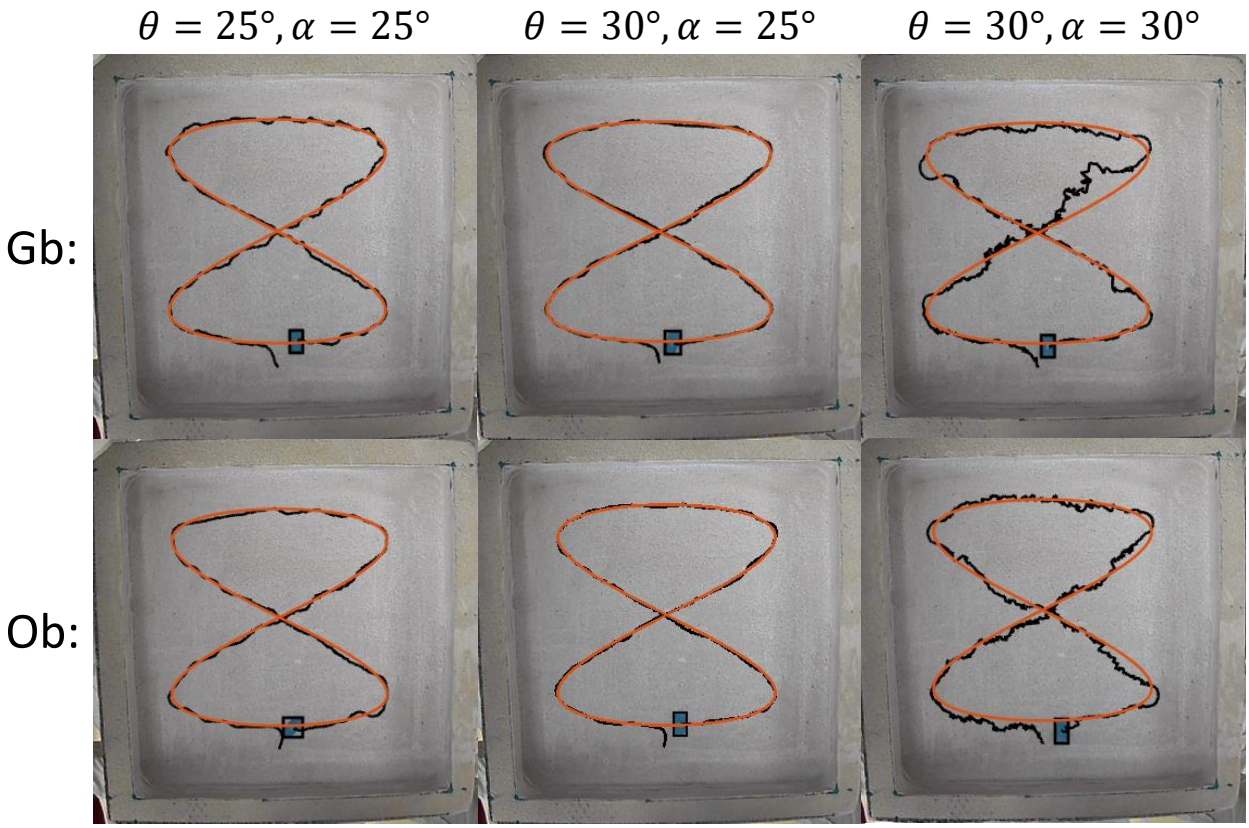


Figure 1.34: Typical trajectory tracking experimental results for different sweep angles and tilt angles, when the time step is constant at $\Delta t = 0.1$s and a comparison between two controllers. The first row of pictures belongs to the geometric based controller denoted by Gb. The second row is for optimization based controller (Ob). The experimental set values for each column is placed at the top. The orange and black lines are the desired trajectory and the actual trajectory of the center of the millirobot respectively.

Chapter 2

Robust Nonlinear Quadratic Gaussian Controller

2.1 Problem description

In this chapter, we propose a new Robust Nonlinear Quadratic Gaussian (RNQG) controller based on State-Dependent Riccati Equation (SDRE) scheme for continuous-time nonlinear systems. Existing controllers do not account for combined noise and disturbance acting on the system. The proposed controller is based on a Lyapunov function and a cost function includes states, inputs, outputs, disturbance, and the noise acting on the system. We express the RNQG control law in the form of a traditional Riccati equation.

Real-time applications of a controller place a high computational burden on system implementation. This is mainly due to the nonlinear and complex form of the cost function. In order to solve this problem, this cost function is approximated by a weighted polynomial. The weights are found by using a least-squares technique and an offline neural network. The approximate cost function is incorporated into the controller by employing a method based on Bellman's principle of optimality. Finally, different examples are used to verify the utility of the proposed control approaches.

2.2 Introduction

Most systems are nonlinear in nature and subject to disturbances and noise. To control these systems, one needs a robust controller to deal with these undesired factors. In this chapter, we focus on robust optimal controllers for nonlinear systems that are subject to disturbances, measurement and process noise. The main challenge of regulating this type of systems is accounting for noise and disturbance simultaneously in the control system design. In addition, on-line implementation of the resulting controllers can be too computationally intensive. For this purpose, we present a Neural Network (NN) approximation to render real time implementable controller.

In a recent article, Gabriel *et al.* [48] presented a robust optimal controller for linear systems subject to disturbances. They, however, did not include the effect of noise in their analysis. Van Parys *et al.* [49] also investigated a robust control of constrained stochastic linear systems in the presence of disturbances. A development of robust optimal controller over a linear system with adjustable uncertainty sets was conducted in [50]. Terra *et al.* [51] proposed an optimal robust recursive regulator for linear discrete-time systems that are subject to parametric uncertainties. In all these studies, underlying systems were linear and only the disturbance effect was considered.

As far as nonlinear systems are concerned, several robust optimal control strategies were developed. Hu *et al.* [52] presented a robust H_∞ output-feedback control strategy for the path following of autonomous ground vehicles in the presence of parameter uncertainties and external disturbances. Yang and He [53] proposed an Adaptive Dynamic Programming (ADP)-based self learning robust optimal control scheme for input-affine continuous-time nonlinear systems with mismatched disturbances. They showed that to solve a Hamilton-Jacobi-Bellman-Isaacs (HJBI) equation, they needed a NN approximator. Satici *et al.* [54] designed a robust L_1-optimal control of a quadrotor unmanned aerial vehicle (UAV). Also, Zhang *et al.* [55] studied a robust controller for uncertain nonlinear systems using ADP. They presented an event-based ADP algorithm by designing the NN weight updating laws. Wang *et al.* [56] presented a novel $H_2 - H_\infty$ State-Dependent Riccati Equation (SDRE) control approach. They demonstrated the efficiency of control design framework for continuous-time nonlinear systems. All these studies deal with nonlinear underlying systems. They do consider effects of disturbances and uncertainties. But, the noise was not taken as a factor in their control system development.

In addition, other types of controllers were applied to linear and nonlinear controllers subject to noise. Bian and Jiang [57] studied a robust optimal control on linear stochastic systems with input-dependent noise. Ma *et al.* [58] investigated a class of nonlinear systems with external noises. They presented a NN-based adaptive robust controller to eliminate the effect of the external noises. Telen *et al.* [59] proposed an approximate robust optimal controller for nonlinear systems subjected to process noise. They showed that their technique outperformed the Kalman filter like techniques.

Most of the previous work considered systems subject to disturbances or noise separately. To the best of our knowledge, there is no previous study considered the combined effect of the disturbance and noise together in optimal robust control of a nonlinear system. Here, we propose a control law that can be applied to systems with noise and disturbance combined [60]. Our proposed control law is based on nonlinear SDRE method. A general Lyapunov function based on system dynamics, inputs, outputs, disturbances, and noise is defined. Then, we derive a general Riccati equation to obtain a robust optimal feedback control law. Next, this general Riccati equation is simplified to obtain an $H_2 - H_\infty$ controller. Yet, implementing the conventional SDRE and our proposed controller in real-time applications requires high computational load [61]. This task becomes especially insurmountable for nonlinear systems. For this reason, we use the HJBI equation and NN weighted updating framework to obtain an approximate solution. In this way, it would be possible to use the proposed control scheme in real time. Our proposed control method can be used in wide range of applications such as robotics, space systems, quadrotor UAVs, mobile vehicles, and leader-follower systems.

We apply the proposed controller to upright stabilization of a fly-wheeled inverted pendulum and a tethered satellite system to de-orbit a space debris. The former studies dealing with the flywheel actuated inverted pendulum systems did not include the effect of external disturbances and sensor noise [30, 47, 62, 63]. In our laboratory studies, we discovered that alignment errors, joint backlash, delay, and sensor noise can significantly affect control performance in flywheel stabilized inverted pendulums. Using numerical simulation, we show that the use of the proposed controller significantly improves system performance.

2.3 Control Design

Developing the SDRE based controller for a nonlinear system includes the following steps [64]:

1. Define a cost function.

2. Express the dynamics in state-dependent coefficient form.

3. Solve the resulting Riccati equation.

4. Obtain the controller by using the traditional closed-loop feedback formula, with all the coefficients depending on the states instead of time.

These steps for the control approach is presented in the next section. The procedure results in a Robust Nonlinear Quadratic Gaussian **(RNQG)** controller. This control law is based on a standard form of Riccati equation and does not increase the computational complexity of the problem.

2.3.1 Robust Nonlinear Quadratic Gaussian

In this section, we focus on controlling the following nonlinear system:

$$\dot{\bar{x}} = f(\bar{x}) + B(\bar{x})\,\bar{u} + F(\bar{x})\,\bar{w} + \bar{v} \tag{2.1}$$

$$\bar{y} = C(\bar{x})\,\bar{x} + D(\bar{x})\,\bar{u} + G(\bar{x})\,\bar{w} + \bar{\epsilon} \tag{2.2}$$

where $\bar{x} \in \mathbb{R}^{n\times 1}$, $\bar{u} \in \mathbb{R}^{m\times 1}$, $\bar{y} \in \mathbb{R}^{r\times 1}$, $\bar{w} \in \mathbb{R}^{q\times 1}$, $\bar{v} \in \mathbb{R}^{n\times 1}$, and $\bar{\epsilon} \in \mathbb{R}^{r\times 1}$ are the state of the system, the input applied to system, the output, the external disturbance acting on the system, the process noise, and the measurement noise vectors, respectively. The notation $R^{n\times 1}$ indicates that the matrix R is a $n \times 1$ matrix.

Based on extended linearization [65], and under the assumption $f(\bar{0}) = \bar{0}$, a continuous nonlinear matrix-valued function $A(\bar{x})$ always exists such that:

$$f(\bar{x}) = A(\bar{x})\,\bar{x} \tag{2.3}$$

So we can rewrite the system dynamics as follows:

$$\dot{\bar{x}} = A(\bar{x})\,\bar{x} + B(\bar{x})\,\bar{u} + F(\bar{x})\,\bar{w} + \bar{v} \tag{2.4}$$

$$\bar{y} = C(\bar{x})\,\bar{x} + D(\bar{x})\,\bar{u} + G(\bar{x})\,\bar{w} + \bar{\epsilon} \tag{2.5}$$

where $A^{n\times n}, B^{n\times m}, C^{r\times n}, D^{r\times m}, F^{n\times q}$, and $G^{r\times q}$ are the known state dependent coefficient matrices. We should note that in Eq. (2.4), the vector $\bar{v}$ is a stochastic process called process

noise. Its mathematical characterization is:

$$E(\bar{v}) = 0 \tag{2.6}$$

where the function $E(.)$ is the expected value [66], and:

$$E(\bar{v}(t)\bar{v}^T(t+\tau)) = Lv(\tau) \tag{2.7}$$

where the matrix $L^{n\times 1}$ is called the intensity matrix of the process noise with the property $L\,L^T > 0$ and $v(\tau)$ is a Dirac delta function [66]. In Eq. (2.5), the vector $\bar{\epsilon}$ is a measurement noise, which can be the noises from sensor's extracted data. It is assumed that:

$$E(\bar{\epsilon}(t)) = 0; \quad E(\bar{\epsilon}(t)\bar{\epsilon}^T(t+\tau)) = Hv(\tau) \tag{2.8}$$

where the matrix $H^{r\times 1}$ is called the intensity matrix of the measurement noise with the property $H, H^T > 0$. It is also assumed that process and measurement noise vectors are uncorrelated [66];

$$E(\bar{v}(t)\bar{\epsilon}^T(t+\tau)) = E(\bar{\epsilon}(t)\bar{v}^T(t+\tau)) = 0 \tag{2.9}$$

Consider a Lyapunov function given by:

$$V = \bar{x}^T P(\bar{x})\,\bar{x} \geq 0 \tag{2.10}$$

where $P(\bar{x})^{n\times n}$ is a positive definite matrix. The corresponding cost function J for this case can be written as:

$$J = V(t_f) + \int_0^{t_f} \left(\bar{x}^T Q(\bar{x})\,\bar{x} + \bar{u}^T R(\bar{x})\,\bar{u} + \bar{y}^T S(\bar{x})\bar{y}\right) dt + \gamma_1^2 \int_0^{t_f} \bar{w}^T\bar{w}\,dt + \gamma_2^2 \int_0^{t_f} v^T v\,dt \tag{2.11}$$

where $Q^{n\times n}$, $R^{m\times m}$, and $S^{r\times r}$ are symmetric positive definite matrices and γ_1, γ_2 are real

numbers. The derivative of cost function can be written as:

$$\dot{J} = \dot{V} + \bar{x}^T Q(\bar{x})\,\bar{x} + \bar{u}^T R(\bar{x})\,\bar{u} + \bar{y}^T S(\bar{x})\bar{y} + \gamma_1^2\,\bar{w}^T\bar{w} + \gamma_2^2\,v^T v \leq 0 \tag{2.12}$$

Our goal is to provide optimality conditions for the determination of a set of state feedback gains $K^{m\times n}$ such that:

$$\bar{u} = K(\bar{x})\,\bar{x} \tag{2.13}$$

and the closed-loop system is stable and a desired cost function (Eq. (2.12)) is minimized. To find the optimal gains (K_o), we need to identify the components of the matrix P in the Lyapunov function. By substituting the Lyapunov function, the system model, output, and control input equations in $\dot{J}$, Eq. (2.12) yields:

$$\bar{x}^T P\,\dot{\bar{x}} + \dot{\bar{x}}^T P\,\bar{x} + \bar{x}^T \dot{P}\,\bar{x} + \bar{x}^T Q\,\bar{x} + \bar{u}^T R\,\bar{u} + \bar{y}^T S\,\bar{y} + \gamma_1^2\,\bar{w}^T\bar{w} + \gamma_2^2\,v^T v \leq 0 \tag{2.14}$$

$$\begin{aligned}\Rightarrow\ & \bar{x}^T P\,[A\,\bar{x} + B\,\bar{u} + F\,\bar{w} + L\,v] + [A\,\bar{x} + B\,\bar{u} + F\,\bar{w} + L\,v]^T\,P\,\bar{x} \\ & + \bar{x}^T \dot{P}\,\bar{x} + \bar{x}^T Q\,\bar{x} + \bar{x}^T K^T R\,K\,\bar{x}\ + [C\,\bar{x} + D\,K\,\bar{x} + G\,\bar{w} + H\,v]^T S \\ & [C\,\bar{x} + D\,K\,\bar{x} + G\,\bar{w} + H\,v] + \gamma_1^2\,\bar{w}^T\bar{w} + \gamma_2^2\,v^T v \leq 0\end{aligned} \tag{2.15}$$

Equation (2.15) can be equivalently rewritten in matrix form as:

$$\bar{\xi}^T \boldsymbol{M} \bar{\xi} \leq 0 \rightarrow \begin{bmatrix}\bar{x} & \bar{w} & v\end{bmatrix} \begin{bmatrix} M_1 & M_2 & M_3 \\ M_2^T & M_4 & M_5 \\ M_3^T & M_5^T & M_6 \end{bmatrix} \begin{bmatrix}\bar{x} \\ \bar{w} \\ v\end{bmatrix} \leq 0 \tag{2.16}$$

$$M_1 = P\,[A + B\,K] + [A + B\,K]^T\,P + \dot{P} + Q + K^T\,R\,K + [C + D\,K]^T S\,[C + D\,K]$$

$$M_2 = P\,F + [C + D\,K]^T S\,G$$

$$M_3 = P\,L + [C + D\,K]^T S\,H$$

$$M_4 = G^T S\,G + \gamma_1^2\,I_6;\ \ M_5 = G^T S\,H$$

$$M_6 = H^T S\,H + \gamma_2^2\,I_1$$

where I_i is the i-dimension identity matrix. We should note that the dimensions of the matrices $\boldsymbol{M}, M_1, M_2, M_3, M_4, M_5$ and M_6 are $(n+q+1)\times(n+q+1)$, $n\times n$, $n\times q$, $n\times 1$, $q\times q$, $q\times 1$, and 1×1, respectively.

Definition. In linear algebra, the Schur complement [67] of a symmetric matrix (N) is defined as:

$$N = \begin{bmatrix} N_1 & N_2 \\ N_2^T & N_3 \end{bmatrix} \tag{2.17}$$

where N_1, N_2, and N_3 are respectively $n\times n$, $n\times m$, and $m\times m$ matrices, and N_3 is invertible.

In addition, the Schur complement of the block N_3 of the matrix N is the $n\times n$ matrix defined by:

$$N/N_3 := N_1 - N_2\, N_3^{-1} N_2^T \tag{2.18}$$

Then, N is negative definite if and only if N and N/N_3 are both negative definite. We use this definition for the matrix $\boldsymbol{M}$ and the matrix blocks N_1, N_2, and N_3 are extracted from Eq. (2.16) as:

$$N_1 = \begin{bmatrix} M_1 & M_2 \\ M_2^T & M_4 \end{bmatrix};\ N_2 = \begin{bmatrix} M_3 \\ M_5 \end{bmatrix};\ N_3 = M_6 \tag{2.19}$$

Thus the Schur complement $Z = N/N_3$ becomes:

$$Z = N_1 - N_2\, N_3^{-1} N_2^T = \begin{bmatrix} Z_1 & Z_2 \\ Z_2^T & Z_3 \end{bmatrix} \tag{2.20}$$

$$\begin{aligned} Z_1 &= M_1 - M_3\, M_6^{-1} M_3^T \\ Z_2 &= M_2 - M_3\, M_6^{-1} M_5^T \\ Z_3 &= M_4 - M_5\, M_6^{-1} M_5^T \end{aligned} \tag{2.21}$$

noting that the matrix $Z^{(n+q)\times(n+q)}$ is symmetric. By applying the Schur complement results

on matrix Z again, we have the following $n \times n$ matrix inequality:

$$M_1 - M_3\, M_6^{-1} M_3^T - [M_2 - M_3\, M_6^{-1} M_5^T][M_4 - M_5\, M_6^{-1} M_5^T]^{-1}[M_2 - M_3\, M_6^{-1} M_5^T]^T \leq 0 \tag{2.22}$$

By substituting Eq. (1.12) in to (2.22), we equivalently have:

$$\begin{aligned}
&[P\,[A + B\,K] + [A + B\,K]^T\,P + Q + K^T\,R\,K + [C + D\,K]^T S\,[C + D\,K]] - \\
&\quad [[P\,L + [C + D\,K]^T S\,H][H^T S\,H + \gamma_2^2\,I_1]^{-1}[P\,L + [C + D\,K]^T S\,H]^T] - [[P\,F + [C + D\,K]^T S\,G] - \\
&\quad [P\,L + [C + D\,K]^T S\,H][H^T S\,H + \gamma_2^2\,I_1]^{-1} H^T S\,G][[G^T S\,G + \gamma_1^2\,I_6] - \\
&\quad G^T S\,H[H^T S\,H + \gamma_2^2\,I_1]^{-1} H^T S\,G]^{-1} \\
&\quad [[P\,F + [C + D\,K]^T S\,G] - [P\,L + [C + D\,K]^T S\,H][H^T S\,H + \gamma_2^2\,I_1]^{-1} H^T S\,G]^T \leq -\dot{P}
\end{aligned} \tag{2.23}$$

In order to guarantee stability, the Lyapunov function must be decreasing ($\dot{V} \leq 0$). We can assume that the derivative of matrix P is negative definite $\dot{P} \leq 0$. Thus, the left hand side of the inequality Eq. (2.23) becomes equal to zero. By grouping the different terms of K in Eq. (2.23), we have:

$$\Gamma_1 + \Gamma_2 K + K^T \Gamma_2^T + K^T \Gamma_3 K = 0 \tag{2.24}$$

$$\begin{aligned}
\Gamma_1 =& P\,A + A^T P + Q + C^T S\,C - (P\,L + C^T S\,H) M_6^{-1} (P\,L + C^T S\,H)^T - \\
& [(P\,L + C^T S\,H) M_6^{-1} H^T S\,G - (PF + C^T S\,G)]\Gamma_4 \\
& [(P\,L + C^T S\,H)^T M_6^{-1} H^T S\,G - (PF + C^T S\,G)]^T \\
\Gamma_2 =& P\,B + C^T S\,D - (P\,L + C^T S\,H) M_6^{-1} H^T S\,D - [(P\,L + C^T S\,H) M_6^{-1} H^T S\,G - (PF + C^T S\,G)] \\
& \Gamma_4\, G^T [I_r - S\,H\,M_6^{-1} H^T] S\,D \\
\Gamma_3 =& R + D^T S\,D - D^T S\,H\,M_6^{-1} H^T S\,D - D^T S[G\,\Gamma_4\, G^T - \\
& G\,\Gamma_4\, G^T S\,H\,M_6^{-1} H^T - H\,M_6^{-1} H^T S\,G\,\Gamma_4\, G^T S + \\
& H\,M_6^{-1} H^T S\,G\,\Gamma_4\, G^T S\,H\,M_6^{-1} H^T] S\,D \\
\Gamma_4 =& [[G^T S\,G + \gamma_1^2\,I_6] - G^T S\,H[H^T S\,H + \gamma_2^2\,I_1]^{-1} H^T S\,G]^{-1}
\end{aligned}$$

By completing the square [68] in gain K and use the optimal gain K_o, we have:

$$\Gamma_1 + (K - K_o)^T \Gamma_3 (K - K_o) - K_o^T \Gamma_3 K_o = 0 \tag{2.25}$$

We should note that the matrices Γ_3 and Γ_3^{-1} are symmetric, so by comparing Eqs. (2.24) and (2.25), the optimal feedback gain should be:

$$K_o = -\Gamma_3^{-1} \Gamma_2^T \tag{2.26}$$

We substitute $K = K_o$ in Eq. (2.25), then the matrix P can be calculated by the solution of the following equation:

$$\begin{aligned}
\Gamma_1 - K_o^T \Gamma_3 K_o = \Gamma_1 - \Gamma_2 \Gamma_3^{-1} \Gamma_2^T = {} & P\,A + A^T P + Q + C^T S\,C - (P\,L + C^T S\,H) M_6^{-1} (P\,L + C^T S\,H)^T - \\
& [(P\,L + C^T S\,H) M_6^{-1} H^T S\,G - (PF + C^T S\,G)]\,\Gamma_4\,[(P\,L + C^T S\,H)^T M_6^{-1} H^T S\,G - (PF + C^T S\,G)]^T - \\
& (P\,B + C^T S\,D - (P\,L + C^T S\,H) M_6^{-1} H^T S\,D - [(P\,L + C^T S\,H) M_6^{-1} H^T S\,G - (PF + C^T S\,G)] \\
& \Gamma_4\,G^T [I_r - S\,H\,M_6^{-1} H^T] S\,D)(R + D^T S\,D - \; D^T S\,H\,M_6^{-1} H^T S\,D - D^T S [G\,\Gamma_4\,G^T - \\
& G\,\Gamma_4\,G^T S\,H\,M_6^{-1} H^T - H\,M_6^{-1} H^T S\,G\,\Gamma_4\,G^T S + H\,M_6^{-1} H^T S\,G\,\Gamma_4\,G^T S\,H\,M_6^{-1} H^T] S\,D)^{-1} \\
& (P\,B + C^T S\,D - (P\,L + C^T S\,H) M_6^{-1} H^T S\,D - [(P\,L + C^T S\,H) M_6^{-1} H^T S\,G - (PF + C^T S\,G)] \\
& \Gamma_4\,G^T [I_r - S\,H\,M_6^{-1} H^T] S\,D)^T = 0
\end{aligned} \tag{2.27}$$

Although this generalized Riccati equation seems quite complicated, we need to rewrite the entire equation in the form of conventional Riccati equation. This algebraic transformation helps to use the standard Riccati solver and find the optimal gain from Eq. (2.26). Following notations are defined to obtain the standard Riccati equation.

$$\lambda_1 = \Gamma_4\, G^T[I_r - S\, H\, M_6^{-1} H^T] S\, D$$

$$\lambda_2 = \Gamma_3^{-1}$$

$$\lambda_3 = L\, M_6^{-1} H^T S\, G - F$$

$$\lambda_4 = C^T S\, H\, M_6^{-1} H^T S\, G - C^T S\, G$$

$$\lambda_5 = C^T S\, D - C^T S\, H\, M_6^{-1} H^T D$$

$$\lambda_6 = B - L\, M_6^{-1} H^T S\, D$$

$$\Lambda_1 = -L\, M_6^{-1} H^T S\, C + \lambda_3\, \Gamma_4\, \lambda_4^T - [\lambda_6 - \lambda_3\, \lambda_1]\, \lambda_2 [\lambda_5 - \lambda_4\, \lambda_1]^T$$

$$\Lambda_2 = Q + C^T S\, C - C^T S\, H\, M_6^{-1} H^T S\, C + \lambda_4\, \Gamma_4 \lambda_4^T - [\lambda_5 - \lambda_4\, \lambda_1]\, \lambda_2 [\lambda_5 - \lambda_4\, \lambda_1]^T$$

$$\Lambda_3 = L\, M_6^{-1} L^T + \lambda_3\, \Gamma_4 \lambda_3^T - [\lambda_6 - \lambda_3\, \lambda_1]\, \lambda_2 [\lambda_6 - \lambda_3 \lambda_1]^T$$

Then the conventional form of generalized Riccati equation can be presented as follows:

$$P(A + \Lambda_1) + (A + \Lambda_1)^T P + \Lambda_2 + P\, \Lambda_3\, P^T = 0 \tag{2.28}$$

If we do not have the noise on the system, that is, $L = H = 0$, and only H_∞ performance criterion exists, then the general controller can be converted to $H_2 - H_\infty$ control [56]. The optimal feedback gain can be found as follows:

$$E_1 = [G^T S\, G + \gamma_1^2\, I_6]^{-1}$$

$$K_o = -\,[R - D^T S\, G\, E_1\, G^T D + D^T S\, D]^{-1} [P\, B + C^T S\, D - (PF + C^T S\, G) E_1\, G^T S\, D]^T \tag{2.29}$$

In addition, the solution of conventional SDRE control can be regulate as follows. The simplest system model and its cost function used in the solution of the benchmark problem

is:

$$\dot{\bar{x}} = A(\bar{x})\,\bar{x} + B(\bar{x})\,\bar{u} \tag{2.30}$$

$$J = \int_0^{t_f} [\bar{x}^T Q(\bar{x})\,\bar{x} + \bar{u}^T R(\bar{x})\,\bar{u}]dt \tag{2.31}$$

The solution of the optimal control to minimize the cost function is obtained by solving the following Riccati equation:

$$A^T(\bar{x})\,P(\bar{x}) + P(\bar{x})\,A(\bar{x}) - P(\bar{x})\,B(\bar{x})\,R^{-1}(\bar{x})\,B^T(\bar{x})\,P(\bar{x}) + Q(\bar{x}) = 0 \tag{2.32}$$

The resulting Riccati equation solution is a function of the states. The optimal feedback controller is:

$$\bar{u} = -R^{-1}(\bar{x})\,B^T(\bar{x})\,P(\bar{x})\,\bar{x} \tag{2.33}$$

The SDRE controller, by its construction, ensures that there is a near optimal solution for the system [69].

In implementing SDRE approach, the most desirable option is to solve Eq. (2.32) in a closed form and using a symbolic software package. This may be possible for some systems having special forms and structures. In general, however, an analytical solution cannot be obtained. In which case, the second option is to obtain numerical solution of the problem in real time at a relatively fast rate. The time increment of the discretized solutions of Eqs. (2.32) and (2.33) can be automatically set by a simple Euler or the Runge-Kutta routine.However, in general, real time implementation of the SDRE is computationally taxing [70]. This is particularity true for high order systems [71]. Thus, we propose an approximation method that can highly speed up the real time computations, which is presented in next subsection.

2.3.1.1 The Approximation Method

We seek an approximation $\widetilde{u}(\bar{x})$ to the SDRE controller such that it is also a solution of the HJBI equation [72]. For this purpose, a positive definite cost function $V(\bar{x})$ is required

such that it satisfies the following equation:

$$\min_{u} \left\{ u^T \frac{R(\bar{x})}{2} u + \nabla V^T(\bar{x}) \dot{\bar{x}} \right\} + \widetilde{Q}(\bar{x}) = 0 \tag{2.34}$$

where ∇V is the partial derivative of function V with respect to $\bar{x}$. We should note that $\widetilde{Q}(\bar{x})$ is a positive definite, which may be differ from $Q(\bar{x})$ in Eq. (2.31). Here, we consider two sets of equations of motion for SDRE and RNQG control systems given by:

$$\dot{\bar{x}} = f(\bar{x}) + g(\bar{x})u \tag{2.35}$$

$$\dot{\bar{x}} = f(\bar{x}) + g(\bar{x})u + F(\bar{x})\,\bar{w} + \bar{v} \tag{2.36}$$

The minimization in Eq. (2.34) can be readily obtained by:

$$\widetilde{u}(\bar{x}) = -R^{-1}(\bar{x}) g(\bar{x}) \nabla V(\bar{x}) \tag{2.37}$$

which when substituted into the HJBI equation to yield:

$$\widetilde{Q}(\bar{x}) - \nabla V^T(\bar{x}) g(\bar{x}) \frac{R^{-1}(\bar{x})}{2} g^T(\bar{x}) \nabla V + \nabla V^T(\bar{x}) f(\bar{x}) = 0 \tag{2.38}$$

We should note that, one can find the approximation of SDRE and RNQG by substituting $\dot{\bar{x}}$ with the corresponding terms in Eqs. (2.35) and (2.36), respectively. The function $\widetilde{Q}(\bar{x})$ is considered as an unknown and the function $R(\bar{x})$ is taken as known. In general, there does not exist a function $V(\bar{x})$ such that SDRE controller (Eq. (2.33)) is expressible as Eq. (2.37). Ideally the approximation problem should be combined with the calculation of $V(\bar{x})$ and $\widetilde{Q}(\bar{x})$ in a single step in Eq. (2.38). This however leads to a difficult problem [73, 74]. A necessary condition that $\widetilde{u}(\bar{x})$ should satisfy is that it should be expressible in the form of Eq. (2.37). Here, the approach is to find a polynomial function $V_0(\bar{x})$, which when substituted for $V(\bar{x})$ in Eq. (2.37), the result gives a controller $\widetilde{u}(\bar{x})$, which is the approximation of controllers in the least square sense. The approximation in the least squares sense is chosen in the interest of simplicity and tractability. The coefficients in the polynomial function $V_0(\bar{x})$ are determined by following a learning process. We should note that the approximation function

should be a quadratic polynomial as follows.

$$\widetilde{V}_0(\bar{x}) = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij} x_i^{2j} \tag{2.39}$$

where N is the number of states and K is the number of selected basis polynomial functions.

To explain the idea of approximation method, the system in Eq. (2.35) is represented in the discrete form and an algorithm is proposed. We express the dynamics of a system in the discrete form as:

$$\bar{x}_{k+1} = f(\bar{x}_k) + g(\bar{x}_k) u_k \tag{2.40}$$

where k is the discrete time variables. The cost function in the discrete time form and its recurrence equations can be written as:

$$J = \Theta(\bar{x}_N) + \sum_{k=0}^{N-1} \Theta(\bar{x}_k) \tag{2.41}$$

$$J = J_N(\bar{x}_N) + \sum_{k=0}^{N-1} J_k(\bar{x}_k) \tag{2.42}$$

where

$$J_N(\bar{x}_N) = \Theta(\bar{x}_N), \tag{2.43}$$

$$J_k(\bar{x}_k) = \Theta(\bar{x}_k) + J_{k+1}(\bar{x}_{k+1}) \tag{2.44}$$

where N denotes the final time. The objective is to approximate the function $J_k(\bar{x}_k)$ as a polynomial of the states. A neural network (NN) as an approximator is trained for this purpose. Moreover, with the function approximation, the cost function for the network can be written as:

$$J_k(\bar{x}_k) \cong W_k^T \Upsilon(\bar{x}_k) \tag{2.45}$$

where W_k is the unknown optimal weights of the network at time step and Υ is the basis

function of states. The training process for weights W_k, is presented in Fig. 2.1. Once the algorithm converges, the cost function is approximated by $W_k^T \Upsilon(\bar{x}_k)$ in a closed form.

Algorithm 1

1: $k \leftarrow N$
2: $W_{N+1}^T = \bar{0}$
3: *loop*:
4: Select random η different states, $\bar{x}_k^{(j)}$, $j \in \{1, 2, \cdots, \eta\}$ for $\eta \gg 1$
5: Train network weights for W_k such that

$$W_k^T \Upsilon(\bar{x}_k^{(j)}) = \Theta(\bar{x}_k^{(j)}) + W_{k+1}^T \Upsilon(f(\bar{x}_k^{(j)}))$$

6: $k \leftarrow k - 1$
7: **go to** *loop*.

Figure 2.1: Offline neural network algorithm to find the approximation polynomial.

One may use the method of least squares to find the unknown weights as coefficients of the selected polynomial in the training process [75, 76]. For this purpose, η random states should be selected to apply in the least squares method. Using the equation for train network weights in Fig. 2.1 yields:

$$\begin{cases} W^T \Upsilon(x^{[1]}) = \nu(x^{[1]}) \\ \vdots \\ W^T \Upsilon(x^{[\eta]}) = \nu(x^{[\eta]}) \end{cases} \tag{2.46}$$

where $\nu(x^{[j]}) = \Theta(\bar{x}^{(j)}) + W^T \Upsilon(f(\bar{x}^{(j)}))$. If we define $\boldsymbol{\Upsilon} \equiv [\Upsilon(x^{[1]}), \Upsilon(x^{[2]}), \cdots, \Upsilon(x^{[\eta]})]$ and $\boldsymbol{\nu} \equiv [\nu(x^{[1]}), \nu(x^{[2]}), \cdots, \nu(x^{[\eta]})]$ and using the method of least squares, the solution of Eq. (2.46) is:

$$\boldsymbol{W} = (\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T)^{-1}\boldsymbol{\Upsilon}\boldsymbol{\nu}^T \tag{2.47}$$

Once the neural network controller trained through Algorithm 1, it can be utilized for real time optimal control of the system. The controller can be implemented in real time by substituting $\bar{x}_k$ at each time step into Eq. (2.34) to calculate $\widetilde{u}$ and applying it to the system. The convergence and the stability of Algorithm 1 can be found in [77].

In next chapters, we discuss different nonlinear systems and apply the proposed controllers and compare the outcomes.

2.4 Nonlinear system Applications

2.4.1 Flywheel based inverted pendulum

2.4.1.1 System description

The inverted pendulum considered here is presented in Fig. 2.2. As shown, the pivot point of the pendulum is at point A. The joint at A is free to rotate and is not actuated. A flywheel is attached to the tip of the pendulum. The flywheel is driven by a motor attached at point B. The control objective is to stabilize the inverted pendulum at its vertical position by using inertial actuation generated by the flywheel. Typically IMU sensors are used to measure the angular position (θ) of the pendulum. Such sensors are often prone to measurement noise.

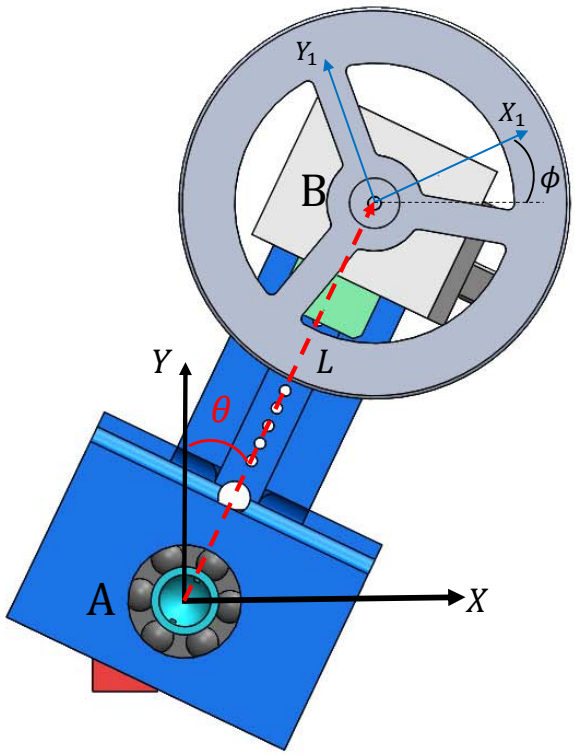


Figure 2.2: Schematics of the flywheel-actuated inverted pendulum.

The mass of the pendulum is M_p and M_w is the pendulum mass. The parameters L_e and L_G are the lengths of the elbow and the distance between the rotation center and the center of gravity of the pendulum, respectively. Figure 2.2 depicts the reference frame XY, and the body frame X_1Y_1, placed at the centers of rotation of the pendulum and the flywheel, respectively. The state variables of the system are $\{\theta, \phi\}$, where θ is pendulum angle and ϕ is the rotation angle of the flywheel in the counter-clockwise direction.

2.4.1.2 Equations of motion

The equations of motion are derived using Lagrange's method (2.48) and can be obtained by substituting the total kinetic (T_{total}) and potential (V_{total}) energies of the system into the Lagrangian ($\mathfrak{L}$) equation.

$$\mathfrak{L} = T_{total} - V_{total};\ \frac{\partial}{\partial t}[\frac{\partial \mathfrak{L}}{\partial \dot{q}_i}] - \frac{\partial \mathfrak{L}}{\partial q_i} = Q_i \tag{2.48}$$

where q_i and Q_i are the i^{th} generalized coordinate and generalized force, respectively. Total kinetic and potential energy expressions of the system can be written as:

$$T_{total} = \frac{1}{2}(M_p L_G^2 + M_w L_e^2 + I_p + I_w)\dot{\theta}^2 + I_w \dot{\theta}\dot{\phi} + \frac{1}{2} I_w \dot{\phi}^2 \tag{2.49}$$

$$V_{total} = (M_p L_G + M_w L_e) g \cos\theta \tag{2.50}$$

where I_p is the moment of inertia of the pendulum and I_w is the moment of the inertia of flywheel. The equations of the motion of the system are given by:

$$(M_p L_G^2 + M_w L_e^2 + I_p + I_w)\ddot{\theta} + I_w \ddot{\phi} - (M_p L_G + M_w L_e) g \sin\theta = 0 \tag{2.51}$$

$$I_w(\ddot{\theta} + \ddot{\phi}) = T_w \tag{2.52}$$

where T_w is the flywheel drive torque as the input to the system. In the next step, to control the flywheel at the desired speed, the mathematical model of the motor's physical behavior is included in the system as follows:

$$V = L_m \frac{di}{dt} + R_m i + K_e \omega_m;\ \ T_w = N_g K_t i \tag{2.53}$$

where V is the motor voltage and i is the armature current. L_m and R_m are the armature coil inductance and resistance, respectively. The motor back electro-magnetic force is given bu K_e and ω_m is the angular velocity of the motor. The gear ratio and the motor torque constant are given by N_g and K_t, respectively. Using the relationship between the motor and the flywheel, we can calculate the required motor voltage in terms of the flywheel angular velocity.

The state-space vector is defined as $\bar{x} = \{\theta, \phi, \dot{\theta}, \dot{\phi}\}$ and the equations of motion is represented in the state-space form as:

$$\dot{\bar{x}} = \bar{F}(\bar{x}) + \bar{G}\, u \tag{2.54}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ \frac{C_T}{I_t} \sin x_1 \\ -\frac{C_T}{I_t} \sin x_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{I_T} \\ \frac{(I_T + I_w)}{I_w I_T} \end{bmatrix} T_w \tag{2.55}$$

$$C_T = (M_p L_G + M_w L_e) g$$

$$I_T = M_p L_G^2 + M_w L_e^2 + I_p$$

2.4.1.3 Simulation results

To test the performance and illustrate the efficiency of the proposed controllers, three simulation examples are carried out using Wolfram Mathematica. First, a comparative simulation among controllers, the SDRE, SDRE approximation, $H_2 - H_\infty$, RNQG, and RNQG approximation methods is presented where the system is considered as an ideal one. In the second example, the performance of the controllers are examined in the presence of disturbance and noise. The system parameters used in the simulations are presented in Table 2.1. The comparisons between controllers are shown in Figs. 2.3-2.5. Also, we should note that the sampling time of the simulation is $\Delta t = 0.01$ s.

Table 2.1: The details of the system's parameters.

parameter	value	parameter	value
M_p	0.6 Kg	L_e	14 cm
M_w	0.31 Kg	I_p	0.0023 Kg/m^3
L_G	10 cm	I_w	0.001 Kg/m^3

2.4.1.4 Ideal System

In the first simulation (Case 1), we consider an ideal system (3.4) and (3.5) with the following parameter matrices and initial conditions:

$$Q = \begin{pmatrix} 1+\theta^2 & 0 & 0 & 0 \\ 0 & 1+\phi^2 & 0 & 0 \\ 0 & 0 & 1+\dot{\theta}^2 & 0 \\ 0 & 0 & 0 & 1+\dot{\phi}^2 \end{pmatrix}; \quad R = I_1;$$

$$S = I_4; \quad \bar{x}_0 = \{20^\circ, 0^\circ, 0.01 \text{ rad/s}, 0 \text{ rad/s}\} \tag{2.56}$$

As can be seen in Fig. 2.3, all controllers are stable and they converge to the desired values. It can be observed that the pendulum stabilizes in the vertically upright position quickly and smoothly after a minor overshoot. In addition, from the analysis of the simulation results, the responses of the control schemes and approximation method are similar as we expected. Note that the tracking error can be remarkably reduced by selecting larger magnitudes for the components of the two matrices Q and R in all control schemes. This also leads to faster convergence speed. In addition, as one can see, the performance of the proposed RNQG is much better than that other controllers.

2.4.1.5 System subjected to disturbance and noise

An external disturbance and zero mean white noise are introduced in these simulation runs. Where the external disturbance is applied after 10 seconds, and the noise is added to states as follows:

$$\bar{x} = \bar{x} + v \tag{2.57}$$

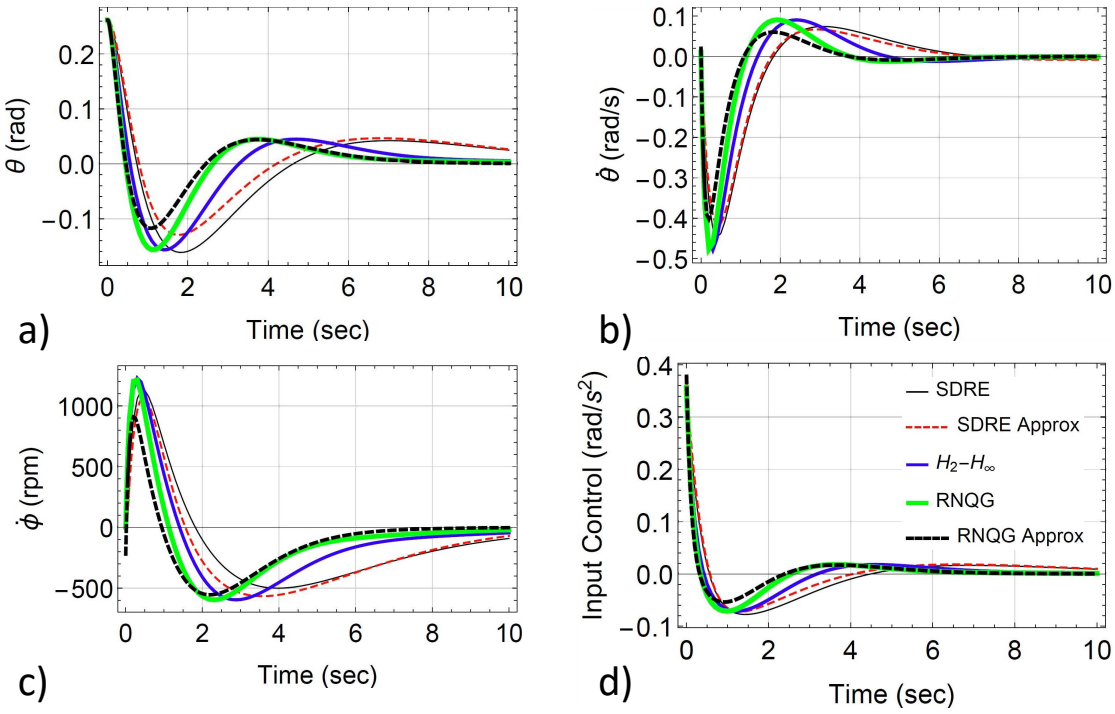


Figure 2.3: Simulation results of system performance in the ideal condition. a) The pendulum angle. b) The pendulum angle rate. c) The flywheel angular velocity. d) The applied input.

where

$$v \sim N(0, q^2) \tag{2.58}$$

We have selected two values 0.04 (Case 2) and 0.4 (Case 3) unit for q. The difference between RNQG and other controllers is more noticeable in this case (see Figs. 2.4 and 2.5). As one can see, the RNQG outperforms the other controllers in presence of noise. The SDRE controller regulates $\dot{\phi}$ smoothly and does not exhibit significant overshoot unlike the RNQG scheme. When, we increased the magnitude of the noise, the performance of the proposed controller was shown better results such as convergence speed and eliminating the noise (see Fig. 2.5). In this case, we only compared the SDRE, $H_2 - H_\infty$, and RNQG to show the difference in their performances.

For more accurate and quantitative comparison of the proposed controller performances,

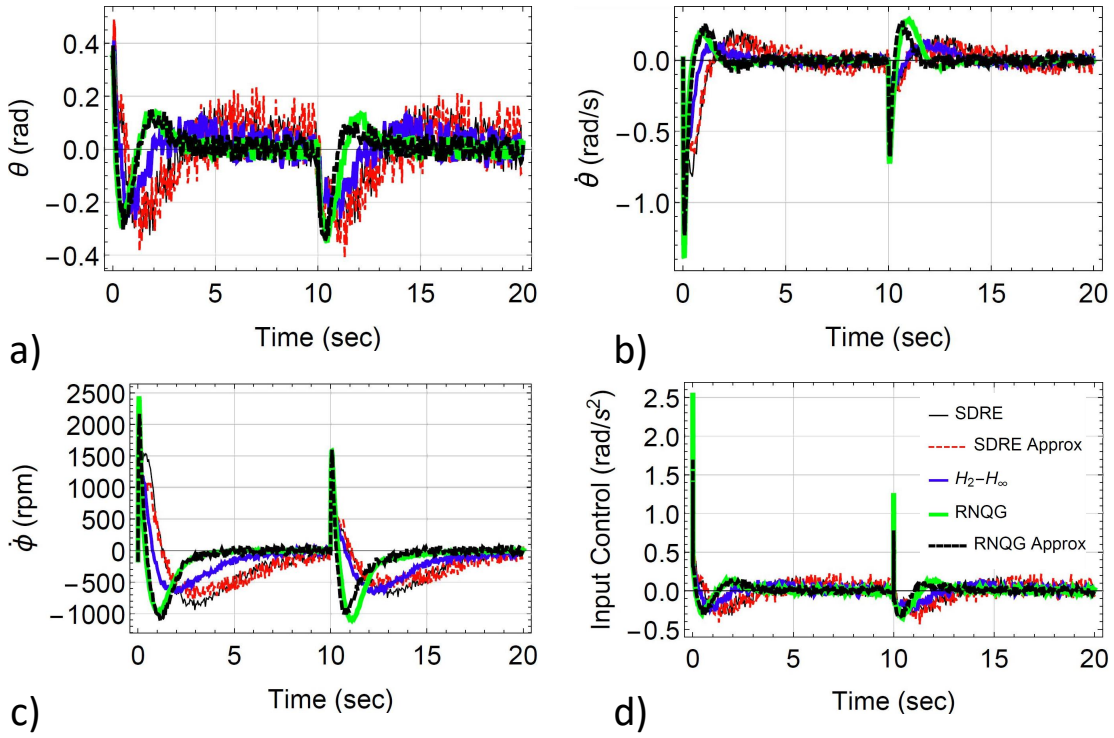


Figure 2.4: Simulation results of system performances in the presence of disturbance and noise. a) The pendulum angle. b) The pendulum angle rate. c) The flywheel angular velocity. d) The applied input.

three performance indicators are considered. These indicator measurements are: Integral Absolute Error (IAE), Integral Time Absolute Error (ITAE), and Control Energy Factor (CEF). They indicate the tracking error performance and the amount of the control effort of the system. These measurements can be calculated as follows:

$$IAE = \sum_{i=1}^{3} \int_0^{t_{sim}} \left| q_i - q_i^d \right| dt \tag{2.59}$$

$$ITAE = \sum_{i=1}^{3} \int_0^{t_{sim}} t \left| q_i - q_i^d \right| dt \tag{2.60}$$

$$CEF = \int_0^{t_{sim}} u^2 dt \tag{2.61}$$

$$\{q_1, q_2, q_3\} = \{\theta, \dot{\theta}, \dot{\phi}\}$$

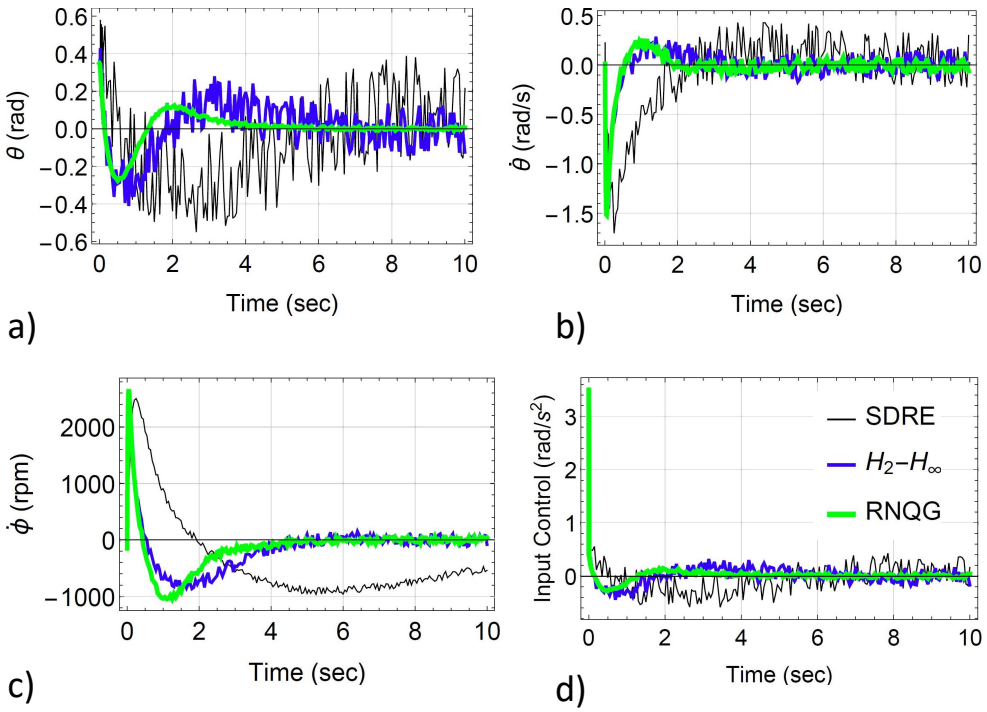


Figure 2.5: Simulation results of system performances in the presence of white noise with 0.4 magnitude. a) The pendulum angle. b) The pendulum angle rate. c) The flywheel angular velocity. d) The applied input.

where t_{sim} is the total simulation time, q is the state, and q^d is the desired value for each state. Table 2.2 presents the performance measurements of the proposed controllers on the system for three cases. In the ideal system, the outcome results showed similar behavior and had just 10 percent improvement. On the contrary, in the presence of noise, the improvement in tracking error became approximately 36 percent.

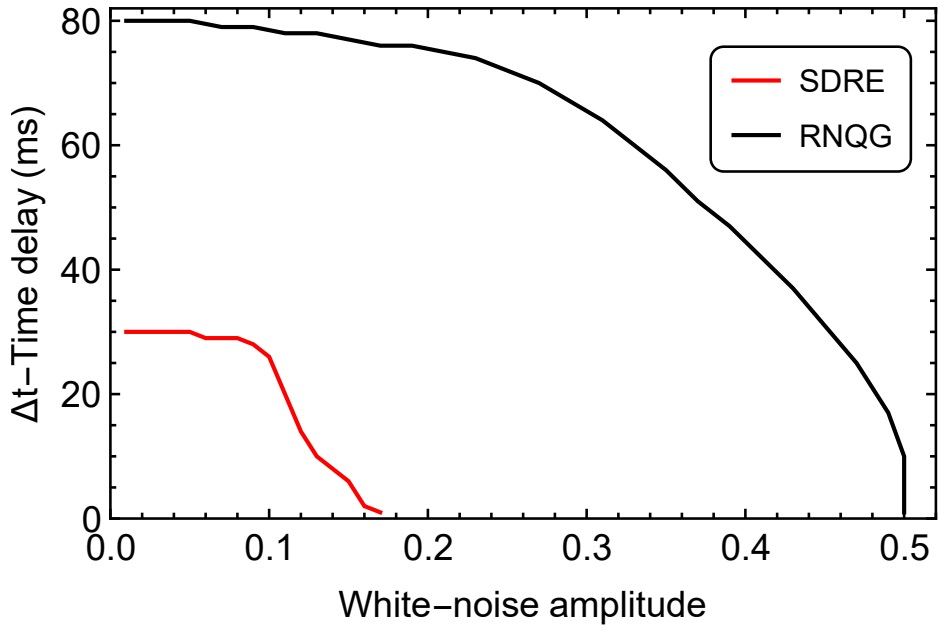
2.4.1.6 Success enclosures of the SDRE and RNQG controllers in the parameter space

In this section, first, we investigate the effect of three parameters on the outcomes of basic SDRE and proposed RNQG control methods. Then, we present and quantitatively compare the effects of varying three parameters on the system behavior. We select the time delay, sensor noise, and uncertainty in the inertia of flywheel as the undesired factors that affect the control performance. We expect that the proposed controller exhibits better performance in eliminating the adverse effects of time delay, sensor noise, and parameter uncertainty. Both

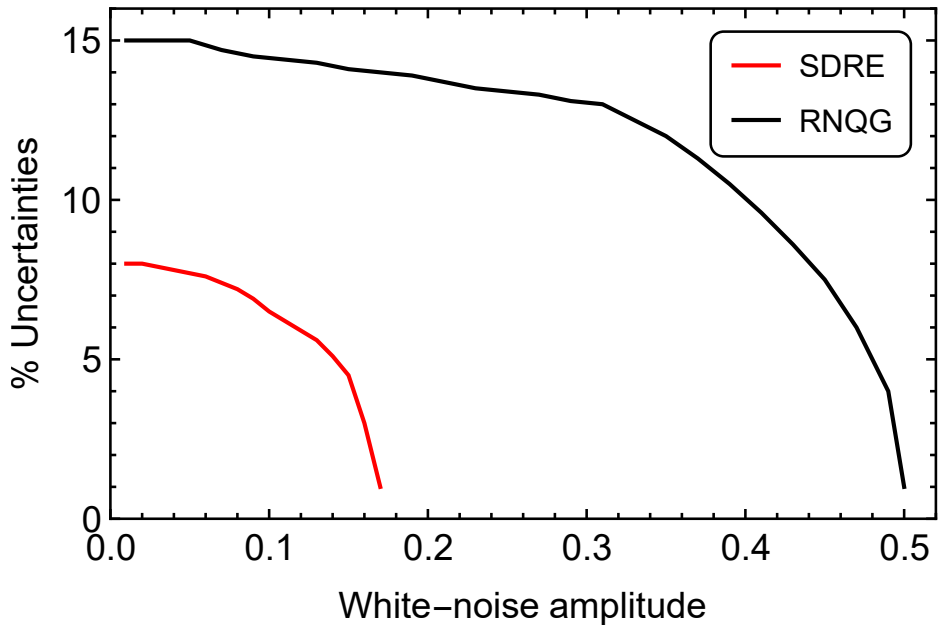
Table 2.2: Performance measurements of the proposed control schemes.

	Controller	IAE	ITAE	CEF
Case1	SDRE	123.03	333.88	0.041
	SDRE Approx.	116.99	305.83	0.038
	$H_2 - H_\infty$	112.01	227.27	0.035
	RNQG	110.21	164.08	0.035
	RNQG Approx.	108.73	158.82	0.033
Case2	SDRE	544.94	2673.56	0.088
	SDRE Approx.	536.51	2655.02	0.084
	$H_2 - H_\infty$	501.48	2320.34	0.054
	RNQG	411.47	1964.52	0.051
	RNQG Approx.	407.83	1899.18	0.049
Case3	SDRE	252.5	299.44	0.094
	$H_2 - H_\infty$	237.69	140.59	0.081
	RNQG	161.97	93.13	0.066

controllers are solved through the use of Riccati equation. During any simulation run, if the Riccati equation fails to converge to a specific solution, we assume that the controller can not stabilize the system. Consequently, the control action at the specific parameter set will be classified as unsuccessful using this identification method. We vary the time delay between 1 to 100 milliseconds by increments of 5 ms, the amplitude of sensor noise between 0.01 to 1 by increments of 0.02, and the percentage of uncertainty between 0 to 20 by increments of 1%. In addition, we use a zero-mean white noise to represent the sensor noise acting on the system. Next, a three dimensional parameter grid is formed, with each grid point representing a triplet of parameter values. Subsequently, two simulation runs (one using SDRE and another using RNQG) are carried out. Using this approach, each grid point with successful control action is identified. This results in two enclosures in the parameter space where individual control methods yield stable response.



(a)



(b)

Figure 2.6: Parametric analysis on the stable solution for different time delays, sensor noise amplitudes, and percentage of uncertainties in the inertia of flywheel. It shows the boundaries between the stable and unstable solutions where the left hand side of the curves belongs to the stable solutions. (a) Constant uncertainty at 1 % (b) Constant time delay at 1 ms

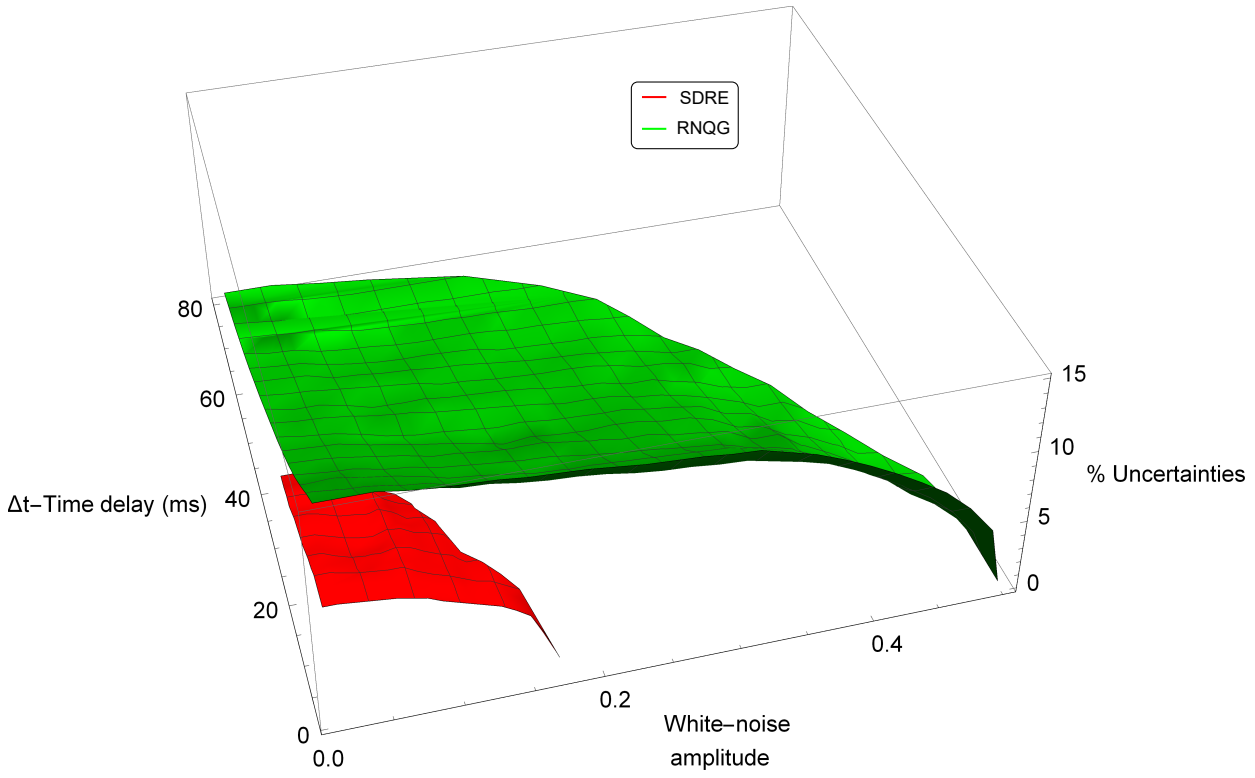


Figure 2.7: Three-Dimensional representation of the two enclosures in the parameter space.

Figure 2.6 represents projections of the results of the parametric analysis onto the Noise-Delay and the Noise-Uncertainty planes. The left hand side of the curves belongs to the stable solutions; however, the solutions would be unstable by choosing any parameters on the right hand side of the curves. The Figure 2.6a shows the variation of the maximum time delay with respect to the amplitude of white noise, while the percentage of uncertainty is fixed at 1 percent, and the system still has a stable solution. Also, Fig. 2.6b depicts the variation of the maximum percentage of uncertainty with respect to the amplitude of white noise when the time delay is fixed at one millisecond. As one can see, the enclosure of the proposed RNQG controller is significantly larger that the enclosure of the SDRE controller. Figure 2.7 shows the two enclosures in the three-dimensional space. One can see that the success volume of the RNQG spans larger portion of the parameter space.

2.4.2 Real time control of tethered satellite systems to de-tumble space debris

2.4.2.1 Problem description

Space debris has become a huge concern for orbital missions that makes remediation a critical and necessary action. Using Tethered Satellite System (TSS) to de-orbit debris is one active method to reduce the population of debris in Low Earth Orbits (LEO). We propose a TSS where a satellite is connected to a large space debris by an elastic tether. This system in LEO is subjected to many different disturbances such as aerodynamic drag, which necessitate a robust control method.

Here, we use the simpler version of RNQG, the $H_2 - H_\infty$ controller. Also, we utilize the approximation method for the $H_2 - H_\infty$ control method. The performance of proposed controllers are evaluated by numerical simulations and the results show a convergence of the states to zero. Also, the control law forces the system to decrease the velocity of the debris in the orbit, thus the altitude of debris' orbit decreases automatically so that atmospheric drag will cause the debris to burn out more rapidly by entering Earth's atmosphere.

2.4.2.2 Introduction

Space debris is one of the biggest problems for space missions that has emerged in recent years. Researchers are investigating solutions to solve this problem. Many innovations for cleaning and gathering the space debris are proposed every day and most of them are still in development phases.

There are many removal debris methods that have been proposed. A promising method is the use of a vehicle to capture the debris and return it to a central station. Missel and Mortari [78] proposed an effective structure capable of capturing and ejecting space debris. This technology captures the debris and, through a series of steps, is able to release the debris in a lower perigee.

Another proposed method includes the use of a laser broom. This method uses a powerful ground-based laser to decrease debris altitude and bring it to a decaying orbit [79–81]. Yet another method is characterized by using a tether to connect debris to a satellite for de-orbit. The Tethered Satellite System (TSS) has a wide range of applications such as tether rendezvous [82], deep space observations [83], and space debris retrieval [84]. This makes

the TSS one of the most active research fields in space sciences. This system includes two or more satellites that are connected together with tether elements. Extensive research has been conducted to study the modeling, dynamics, and control of this type of systems [85,86].

Application in space debris removal has started to be extensively researched as a potential method [87]. Meng *et. al* [88] presented a space debris removal method by using a flexible net and maneuverable units. They established a controller combining the optimization pseudo-dynamics inversion and sliding mode control. Benvenuto *et. al* [89] demonstrated that the "tethered-net system is a promising technology to capture and remove space debris". Nishida *et. al* [90] have investigated a micro-satellite system to remove debris by implementing the use of an electro-dynamic tether (EDT) technology. Aslanov and his coworkers [84,91,92] considered a tethered orbital transfer vehicle in an earlier study. Also, they has investigated the effect of different parameters on such a system. They evaluated "the mutual influence of the tether vibrations and the vibrations of flexible appendages during thrusting phase" and provided a model for the space tug experienced by the debris. Hoyt and Forward [93] patented the terminator tether, which is a lightweight, low-cost device. A conducting tether was used to generate an electrodynamic drag and remove satellite from LEO. They were able to achieve effective results and demonstrated the potential for tethers to be widely used in the removal of space debris. Shan *et. al* [94] studied the deployment dynamic characteristics of tethered-net in the debris capture phase. Also, Zhao *et. al* [95] investigated the contact and post-contact net-capture dynamics of a tethered net, while a space debris was wrapped in a symmetric configuration. Zhang *et. al* [96,97] proposed a maneuverable tethered space net robot presented an adaptation control scheme to symmetrically and asymmetrically capture and release a space debris. Qi *et. al* [98] studied the dynamics and control of a double-tethered space-tug system to remove a space debris. A simple feedback control law was used to regulate the attitude of the debris in a nominal tether length.

Here, we propose an approach to control a nonlinear tether satellite system subject to an external disturbance to detumble, tow, and de-orbit a space debris [99]. In this example, we consider the atmospheric drag as a disturbance acting on the system. We will simulate the proposed system in an orbital maneuver with and without disturbance term in the control scheme and compare the results. It will be shown that the robust controller regulates the

system faster and also reduces the tensile force in the tether tremendously.

2.4.2.3 System Description and Equations of Motion

The system can be described as a Tethered-Satellite-Debris (TSD) system. A massive and non-functional space debris is considered as a point mass that is connected to an active satellite with an elastic tether. The satellite is also treated as a point mass and is placed in a lower altitude orbit than the debris. The satellite is equipped with three continuously operating thrusters. The tether is ensured to remain fully stretched during the control operation, thus its tension will always be positive. We will show that this system will significantly shorten the natural orbital period of the debris. First, the control law de-tumbles the system and regulates it to be placed in the azimuth direction. Then, the thrusters on the satellite are used to decrease the velocity of the system. As a result, the system descends to an atmospheric orbit. The dense atmosphere provides sufficient thermal flux to burn of the debris. Also, the proposed controller ensures maintaining the tensile force in the tether during the operation and produces less force than the conventional controllers to increase the reliability of the system.

2.4.2.4 Dynamical Model

Two connected masses by a uniform fully-stretched tether are rotating around the earth. Two reference frames (the local horizon reference frame (RSW) and the body frame ($b_1b_2b_3$)) placed at the center of mass (COM) of the system is used to derive the equations of motion. Figure. 2.8 shows the schematic representation of the system and the position of the reference frames.

We should note that a common equilibrium condition for this system happens when two masses and the center of the Earth place in a straight line. In this configuration, as one can see in the Fig. 2.8, the body frame is coincided with the RSW frame. The equilibrium conditions are discovered in Section 5.

In Fig. 2.8, variables G, m_d, m_s, r, r_s, and r_d denote the COM, the mass of the debris and satellite, position vectors of TSD COM, satellite and debris, respectively. The angles θ (in-plane) and ϕ (out of plane), the liberation angles, and the length of tether (l) are considered as the states of the system. The tether is assumed to be elastic, and capable of

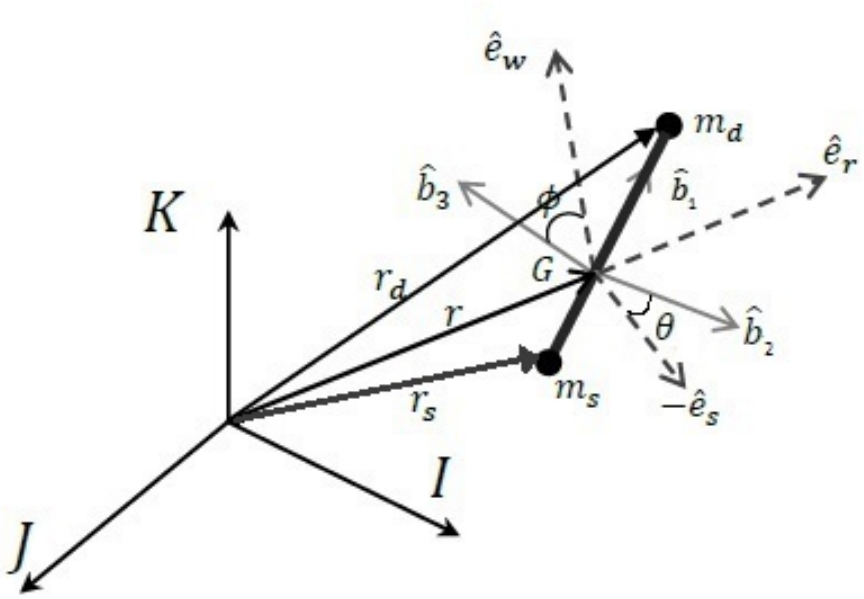


Figure 2.8: Schematic of TSD with the reference frames and respective definition of the liberation angles.

exerting force only along the straight-line connecting the end masses. The objective of the control output is that the in-plane (θ) and out-of-plane (ϕ) libration angles must be zero. We use the Lagrangian method to obtain the equations of motion of this system. By defining the Lagrangian function as $\mathfrak{L} = T_{total} - U_{total}$, one can write the general form of the Lagrange's equation as follows:

$$\frac{\partial}{\partial t}[\frac{\partial \mathfrak{L}}{\partial \dot{\sigma}}] - \frac{\partial \mathfrak{L}}{\partial \sigma} = Q_\sigma, \quad \sigma = \{\theta, \phi, l\} \tag{2.62}$$

where Q_σ is the non-conservative forces, known as the generalized force, and σ are the generalized coordinates. Derivation of the equations of motion by using the Lagrangian approach required to obtain the expressions of kinetic energy (T_{total}), potential energy (U_{total}), and the generalized aerodynamic drag forces [100]. The kinetic and potential energies are presented in Appendix. The drag force acting on the system is presented in the next section.

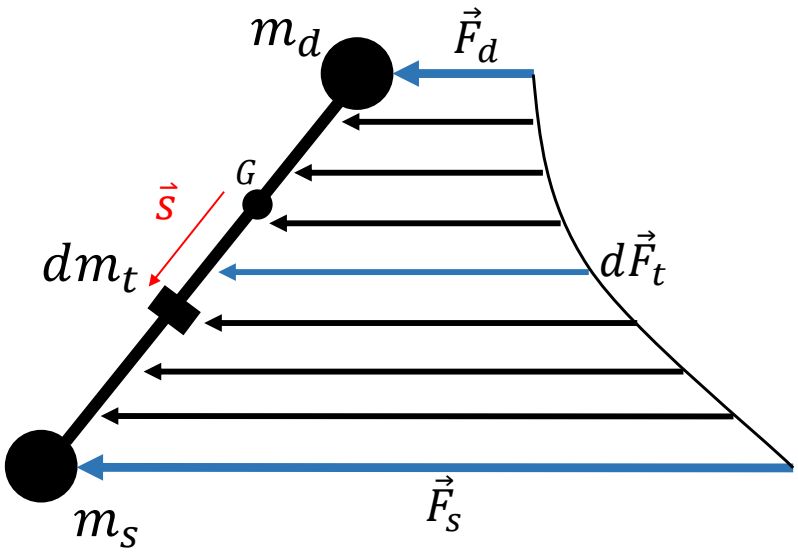


Figure 2.9: Tether mass element along the length of tether and drag forces acting on the satellite, debris, and the tether.

2.4.2.5 Aerodynamic Drag

We consider the aerodynamic drag force as a non-conservative generalized force. The drag force cannot be neglected in LEO. The molecular mean free path of the atmosphere is large in LEO; thus, we use a free-molecular flow model, with the drag coefficient of 2 ($C_D = 2$) for the spherical geometry of the satellite and debris [101]. The drag force vector in the opposite direction of the velocity vector of the system can be given by:

$$\mathbf{F}_i = -\frac{1}{2} C_D \, A_i \, \rho_i V_i \mathbf{V}_i, \quad i = \{s, d\} \tag{2.63}$$

where $\mathbf{F}_i$, A_i, ρ, and $\mathbf{V}_i$ denote the drag force vector acting on each end-mass, the projected area of each end-mass, the air density, and the velocity vector of each end-mass, respectively. It should be noted that the center of mass speed in LEO is more than the relative velocity of each end-mass with respect to the center of mass and it can be neglected. Air density is

modeled as exponentially decaying function in the atmosphere [102]:

$$\rho_i(h) \;=\; \rho_0 \exp(-\frac{\Delta h_i}{H}) \tag{2.64}$$

where ρ_0 is the air density of the center of mass, H is the scale height, and Δh_i is the altitude of each end-mass related to the center of mass.

Similarly, the drag force on the tether can be calculated as follows:

$$d\mathbf{F}_t \;=\; -\frac{1}{2}\, C_{D_t}\, dA_t\, \rho(h_s) V_{dm_t} \mathbf{V}_{dm_t}. \;; \quad dA_t \;=\; \cos\alpha\, d_t\, dl_t \tag{2.65}$$

where V_{dm_t}, dA_t, $\rho(h_s)$, α, and C_{D_t} are the velocity of the tether element, the elemental surface of the tether projected onto the velocity direction, the air density along the tether direction, the angle between velocity and the surface normal direction, and the drag coefficient of the tether, respectively. The tether is modeled as a cylinder with 2.2 drag coefficient ($C_{D_t} = 2.2$) [101], and d_t is the tether diameter.

The right land side of the Lagrangian equation Q_{F_σ} can be calculated using:

$$Q_{F_\sigma} = \mathbf{F}_s.\frac{\partial \mathbf{V}_s}{\partial \dot{\sigma}} + \mathbf{F}_d.\frac{\partial \mathbf{V}_d}{\partial \dot{\sigma}} + \mathbf{M}_t.\frac{\partial \mathbf{\Omega}}{\partial \dot{\sigma}} \quad ; \quad \sigma = \{\theta, \phi, l\} \tag{2.66}$$

where $\mathbf{F}$ and $\mathbf{M}$ are the total non-conservative aerodynamic forces and moment of the two masses and tether (Fig. 2.9). The moment of the aerodynamic force on the tether is calculated as:

$$\mathbf{M}_t = \int \mathbf{s} \times d\mathbf{F}_t \tag{2.67}$$

2.4.2.6 Tether Force

The tethers are assumed to be extensible and capable of exerting force only along the straight-line connecting the respective masses. The tether does not support compression nor any components of shear forces or bending moments and is therefore assumed to be perfectly flexible. The tether force is obtained assuming a Kelvin-Voigt constitutive law; that is, the

tensile stress τ_t, and the longitudinal strain ϵ are given by:

$$\tau_t = E(\epsilon + c\dot{\epsilon})\,; \quad \epsilon = \frac{l - l_0}{l} \tag{2.68}$$

where E is the effective Young's modulus, c is a dissipation constant, l and l_0 are the actual and the unstrained tether length, respectively. Hence, if A is the effective cross-sectional area of the tether, then the tension is given by:

$$f_t = \begin{cases} A\,\tau_t & \epsilon \geq 0 \\ 0 & \epsilon < 0 \end{cases} \tag{2.69}$$

which introduces discontinuities since the tether does not support compression. This force is considered as a condition that should be always positive, otherwise the system fails.

2.4.2.7 Truster Force

Three thrusters are located on the satellite, where each of them is aligned along one of the body frame axes. The force of the thruster is considered as non-conservative and the torque Q_{T_σ} can be computed as:

$$Q_{T_\sigma} = \mathbf{T}_1.\frac{\partial \mathbf{V}_s}{\partial \dot{\sigma}} + \mathbf{T}_2.\frac{\partial \mathbf{V}_s}{\partial \dot{\sigma}} + \mathbf{T}_3.\frac{\partial \mathbf{V}_s}{\partial \dot{\sigma}} \quad ; \qquad \sigma : \{\theta, \phi, l\} \tag{2.70}$$

where $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$ are the thruster forces and are considered as inputs in our control scheme.

2.4.2.8 Equations of Motion

The equations of motion are determined by substituting the total kinetic and potential energies presented in Appendix into the Lagrange's equation (Eq. 2.62) which yields:

$$\frac{1}{6}l\cos\phi\left(2M_1\left(2(\dot{\theta}+\omega_0)(\dot{l}\cos\phi-l\dot{\phi}\sin\phi)+l\ddot{\theta}\cos\phi\right)+\frac{3\mu_g\,l}{r^3}(U_0+U_{J_2})\sin 2\theta\cos\phi M_1\right.$$
$$\left.-\frac{3\sin\theta M_2(r^2\ddot{r}+\mu_g(U_0+U_{J_2}))}{r^2}+3\omega_0\dot{r}\cos\theta M_2\right)=Q_{F_\theta}+Q_{T_\theta} \tag{2.71}$$

$$\frac{1}{6}l\left(2\left(2\dot{l}\dot{\phi}+l\ddot{\phi}\right)M_1+\frac{l\sin 2\phi M_1\left(r^3(\dot{\theta}+\omega_0)^2+3\mu_g(U_0+U_{J_2})\cos^2\theta\right)}{r^3}\right.$$
$$\left.-\frac{3\sin\phi M_2\left(r^2\left(\ddot{r}\cos\theta+\omega_0\dot{r}\sin\theta\right)+\mu_g(U_0+U_{J_2})\cos\theta\right)}{r^2}\right)=Q_{F_\phi}+Q_{T_\phi} \tag{2.72}$$

$$\frac{1}{6}\left(2\ddot{l}M_1+2l\left(\frac{3\mu_g(U_0+U_{J_2})\cos^2\theta\cos^2\phi M_1}{r^3}-M_1\left(\cos^2\phi(\dot{\theta}+\omega_0)^2+\dot{\phi}^2\right)\right)\right.$$
$$\left.+\frac{3\cos\phi M_2\left(r^2(\ddot{r}\cos\theta+\omega_0\dot{r}\sin\theta)+\mu_g(U_0+U_{J_2})\cos\theta\right)}{r^2}\right)=Q_{F_l}+Q_{T_l} \tag{2.73}$$

$$Q_{F_\theta}=D_1\cos\theta\cos\phi\,e^{-\frac{M_d\,l\cos\theta\cos\phi}{H}}\left(M_s\,A_s\,e^{\frac{(M_s+M_d)\,l\cos\theta\cos\phi}{H}}-M_d\,A_d\right)$$
$$-D_2\left(e^{\frac{M_s\,l\cos\theta\cos\phi}{H}}\left(H\sec\theta\sec\phi-M_s\,l\right)-e^{-\frac{M_d\,l\cos\theta\cos\phi}{H}}\left(H\sec\theta\sec\phi+M_d\,l\right)\right)$$
$$Q_{F_\phi}=D_1\sin\theta\sin\phi\,e^{-\frac{M_d\,l\cos\theta\cos\phi}{H}}(-M_s\,A_s\,e^{\frac{(M_d+M_s)l\cos\theta\cos\phi}{H}}+M_d\,A_d)$$
$$+D_2\tan\theta\tan\phi(e^{\frac{M_s\,l\cos\theta\cos\phi}{H}}(H\sec\theta\sec\phi-M_s\,l)-e^{-\frac{M_d l\cos\theta\cos\phi}{H}}(H\sec\theta\sec\phi+M_d\,l))$$
$$Q_{F_l}=D_1\cos\theta\sin\phi\,e^{-\frac{M_d\,\cos\theta\cos\phi}{H}}(M_s\,A_s\,e^{\frac{(M_s+M_d)\,\cos\theta\cos\phi}{H}}+M_d\,A_d)$$
$$-D_2\tan\phi\,(e^{\frac{M_s\,\cos\theta\cos\phi}{H}}(H\sec\theta\sec\phi+M_s\,l)+e^{-\frac{M_d\,l\cos\theta\cos\phi}{H}}(H\sec\theta\sec\phi-M_d\,l))$$
$$Q_{T_\theta}=-M_s\,T_2\,l\,\cos\phi\,;\qquad Q_{T_\phi}=-M_s\,T_3\,l\,;\qquad Q_{T_l}=-M_s\,T_1$$

$$M_d=\frac{m_s+m_t/2}{m_s+m_d+m_t}\,;\qquad M_s=\frac{m_d+m_t/2}{m_s+m_d+m_t}$$
$$M_1=3m_dM_d^2+m_t(M_d^3+M_s^3)+3m_sM_s^2\,;\qquad M_2=2m_dM_d+m_t(M_d^2-M_s^2)-2m_sM_s$$
$$D_1=\frac{1}{2}\rho(h)\,V_G^2\,C_D\,l\,;\qquad D_2=\frac{1}{2}H\,\rho(h)\,V_G^2\cos\alpha\,d_t\,C_{D_t}\,;\qquad U_0=-1$$
$$U_{J_2}=\left(\frac{R_E}{r}\right)^2\frac{J_2}{2}(3\sin^2\psi-1)\,;\qquad \rho(h)=\rho_0e^{\frac{-h_G}{H}}\,;\qquad \cos\alpha=\sqrt{1-\cos^2\phi\,\sin^2\theta}$$

where r, μ_g, ψ, R_E, and ω_0 are the distance between the COM and the center of the Earth, the gravitational parameter of the Earth, the latitude of r, the radius of the Earth, and the angular velocity of RSW, respectively.

2.4.2.9 System mathematical description

Here, the nonlinear system is formulated as:

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}) + \mathbf{B}(\mathbf{q})\,\mathbf{u} + \mathbf{F}(\mathbf{q})\,\mathbf{w} \tag{2.74}$$

$$\mathbf{y} = \mathbf{C}(\mathbf{q})\,\mathbf{q} + \mathbf{D}(\mathbf{q})\,\mathbf{u} + \mathbf{G}(\mathbf{q})\,\mathbf{w} \tag{2.75}$$

where $\mathbf{q} \in \mathbb{R}^{n\times 1}$, $\mathbf{u} \in \mathbb{R}^{m\times 1}$, $\mathbf{y} \in \mathbb{R}^{r\times 1}$, and $\mathbf{w} \in \mathbb{R}^{p\times 1}$ are the state of the system, the control input, the output, and the external disturbance acting on the system, respectively. Also, $\mathbf{A}^{n\times n}, \mathbf{B}^{n\times m}, \mathbf{C}^{r\times n}, \mathbf{D}^{r\times m}, \mathbf{F}^{n\times p}$, and $\mathbf{G}^{r\times p}$ are known state dependent coefficient matrices. The controller used here is the simple form of RNQG formulated in Eq. (2.29) [60].

2.5 Equilibrium and Stability

The equilibrium points, $\tilde{\mathbf{q}}$ can be found by setting $\dot{\mathbf{q}} = \mathbf{0}$ in the equations of motion. We conduct the simulation in a sun-synchronous circular orbit. The system and orbital parameters are listed in Table 2.3. The equilibrium conditions for the libration angles are calculated and presented in Table 2.4.

Table 2.3: The orbital and system parameters of TSD.

Parameter	Values	Parameter	Values
ρ_0 (air density of COM)	1.454×10^{-13} $\frac{\text{Kg}}{\text{m}^3}$	m_s (mass of satellite)	500 Kg
h (orbit height)	600 Km	m_d (mass of debris)	3000 Kg
H (scale height)	71.835 Km	m_t (mass of tether)	10 Kg
i (orbit inclination)	97.8 °	A_s (satellite proj. area)	1 m^2
l (tether length)	0.1 Km	A_d (debris proj. area)	5 m^2
ω_0 (orbital mean motion)	0.001083 $\frac{\text{rad}}{\text{s}}$	d_t (diameter of tether)	2 mm
E (tether Young's modulus)	100 GPa	c (tether dissipation constant)	0.1

Table 2.4: Equilibrium conditions of the libration angles of TSD in a sample orbit.

Angles	θ	ϕ
Condition 1	$\frac{\pi}{2}, \frac{3\pi}{2}$	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$
Condition 2	$0, \pi$	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$

Table 2.5: Eigenvalues of the different equilibrium conditions of TSD.

	θ	ϕ	Eigenvalues
Condition 1	$\frac{\pi}{2}, \frac{3\pi}{2}$	$0, \pi$	$\pm 0.00187691, \pm 0.00153214$
	$\frac{\pi}{2}, \frac{3\pi}{2}$	$\frac{\pi}{2}, \frac{3\pi}{2}$	± 0.00226631 i, ± 0.00126893
Condition2	$0, \pi$	$0, \pi$	± 0.00187691, ± 0.00108412 i
	$0, \pi$	$\frac{\pi}{2}, \frac{3\pi}{2}$	$0, \pm 0.00187776$ i

Defining the state vectors as $\mathbf{q} = \{q_1, q_2, q_3, q_4\} = \{\theta, \dot{\theta}, \phi, \dot{\phi}\}$ and the equations of motion can be stated as:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\theta} & \ddot{\theta} & \dot{\phi} & \ddot{\phi} \end{bmatrix}^T = \begin{bmatrix} q_3 & \mathbf{H}(\mathbf{q}) & q_4 & \mathbf{L}(\mathbf{q}) \end{bmatrix}^T \tag{2.76}$$

where $\mathbf{H}(\mathbf{q})$ and $\mathbf{L}(\mathbf{q})$ can be found from equations of motion. The Jacobain matrix ($\mathbf{J}^*$) in Eq. (2.77) is established for the equilibrium conditions and then the stability of them is examined by *Poincaré-Lyapunov* method. If the real part of the eigenvalues of $\mathbf{J}^*$ are negative, then the equilibrium point is locally asymptotically stable. Otherwise, for positive real part eigenvalue, the equilibrium point is unstable. For zero real part eigenvalues, the point is neutrally stable. If the eigenvalue has only imaginary part, the response is periodic and can be considered stable, because its magnitude does not diverge. The Jacobain matrix can be computed as:

$$\mathbf{J}^* = \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}|_{\tilde{\mathbf{q}}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial \mathbf{H}}{\partial \theta} & \frac{\partial \mathbf{H}}{\partial \dot{\theta}} & \frac{\partial \mathbf{H}}{\partial \phi} & \frac{\partial \mathbf{H}}{\partial \dot{\phi}} \\ 0 & 0 & 0 & 1 \\ \frac{\partial \mathbf{L}}{\partial \theta} & \frac{\partial \mathbf{L}}{\partial \dot{\theta}} & \frac{\partial \mathbf{L}}{\partial \phi} & \frac{\partial \mathbf{L}}{\partial \dot{\phi}} \end{bmatrix} \tag{2.77}$$

The eigenvalues for the sample orbit and typical TSD parameters (presented in Table 2.3) are shown in Table 2.5 for all equilibrium conditions.

The results of Table 2.5 show that in condition 1, the angles correspond to two categories of the eigenvalues. Two pairs of real positive/negative eigenvalues and one imaginary pair with a pair of real positive/negative ones show that the condition 1 is an unstable situation. In condition 2, the angles are also responsible for two pairs of imaginary ones and a zero eigenvalue with a pair of imaginary ones which are represented at stable positions for angles.

2.5.0.1 Results

Simulation analysis carried out using Wolfram Mathematica software in order to test the performance of the presented controllers. The system parameters presented in Table 2.3 are used. The control objective is to regulate the angles and bring the debris to a lower orbit. In order to obtain the required $\tilde{\mathbf{u}}(\mathbf{q})$, a least square approximation of $V(\mathbf{q})$ is used. The least square approximation relies on how to choose random values for $\mathbf{q}$ uniformly distributed in the neighborhood equilibrium point. The basis functions are selected as polynomials x^{2j} ; $j \in \{1, 2, \cdots, 6\}$. One can adjust the accuracy of the approximation by selecting different order of the polynomials. We employ 500 random states at each time step from $\Omega = [-0.2, 0.2]$ (rad) in order to train the network.

2.5.0.2 Robustness

We run the first simulation case to show the robust performance of the proposed controller. We check the performances of two controllers on a disturbed system. In the first controller, we consider the robustness term, but in the second one, the robustness term is eliminated.

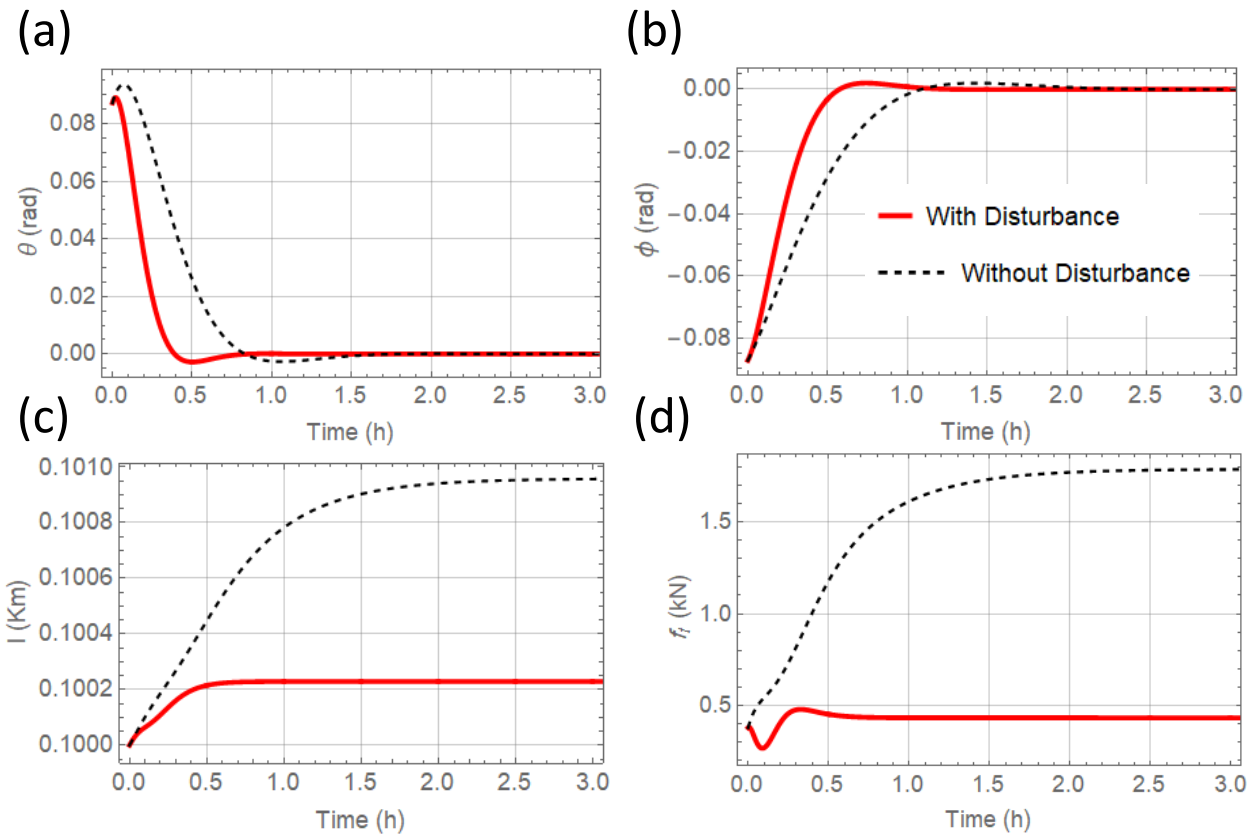


Figure 2.10: Comparison of the performance of two controllers on a disturbed system. The initial conditions are $\mathbf{q_0} = \{0.0872(\text{rad}), 0.2(\text{rad/h}), -0.0872(\text{rad}), 0.1(\text{rad/h})\}$ (a) The in-plane angle. (b) The out plane angle. (c) The length of the tether. (d) The tension existed in the tether.

As one can see from Fig. 2.10, there is not much difference in regulating in-plane and out-of-plane angles, but the controller with disturbance terms significantly reduces the tensile force in the tether.

2.5.1 Case study

In this simulation case, we compare the performance of the controllers and their corresponding approximations. As shown in Fig. 2.11, the thin, thick, thin dashed, and thick dashed lines represent SDRE, $H_2 - H_\infty$, SDRE Approximation, and $H_2 - H_\infty$ Approximation respectively. Comparing controllers and their approximations reveals that the approximation has provided a near optimal solution and zero tracking errors for libration angles are

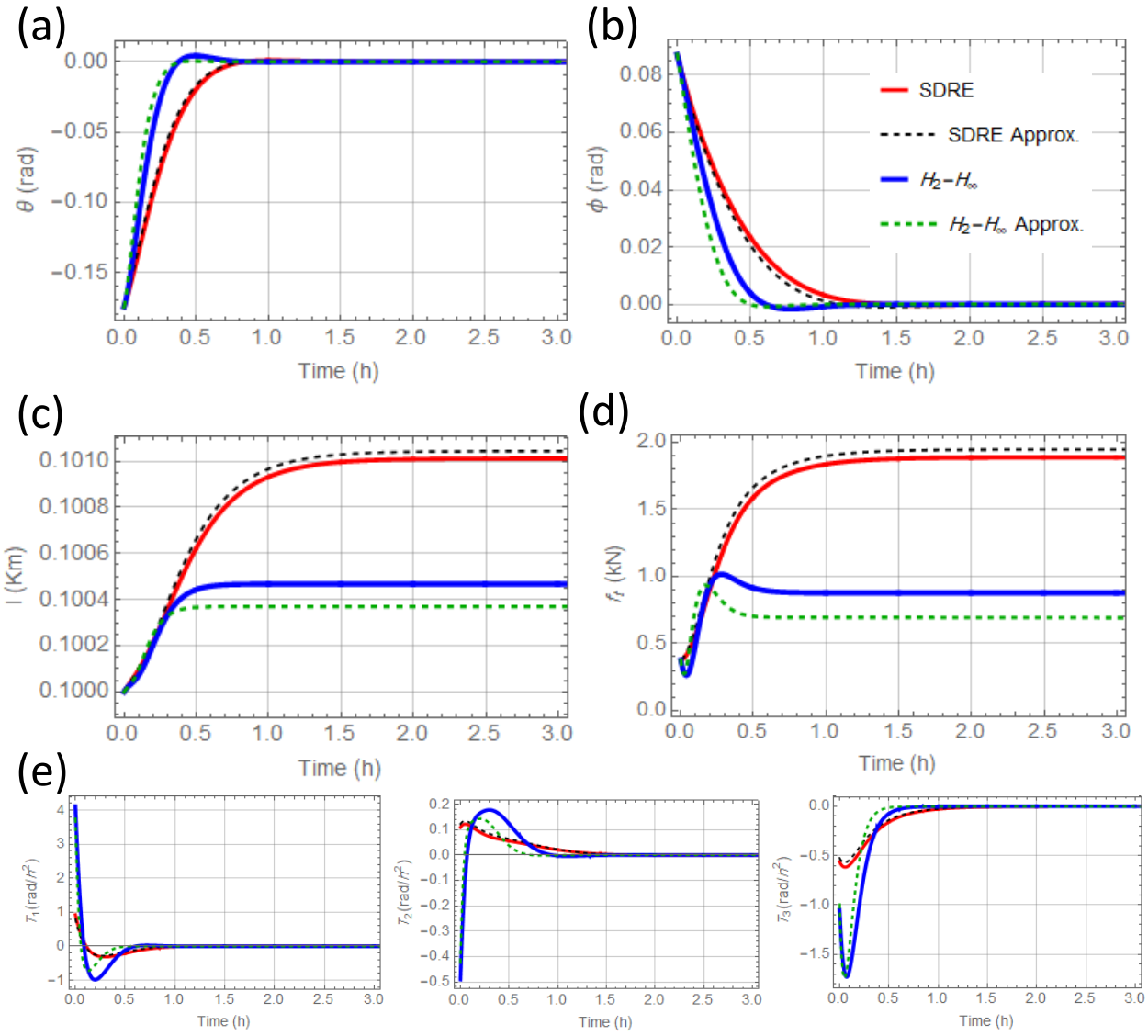


Figure 2.11: Simulation results of the proposed controller and the approximation method. The initial conditions in this case are $\mathbf{q_0} = \{-0.1744(\text{rad}), 0.3(\text{rad/h}), 0.0872(\text{rad}), -0.2(\text{rad/h})\}$. (a) The in-plane angle. (b) The out plane angle. (c) The length of the tether. (d) The tension existed in the tether. (e) The control inputs applied on the system

achieved in all cases. The approximated in-plane and out-of-plane angles follow closely the corresponding controller outputs. Also, as it is shown in Fig. 2.11(d), the controllers force the system to generate positive tensile forces in the tether. That prevents the tether to be slack. In addition, the proposed controller generates less force than the SDRE method.

This gives more reliability to the system that the tether not to be apart. As one can see in Fig. 2.11(e), $\widetilde{\mathbf{u}}(\mathbf{q})$ approximates the corresponding inputs well.

We quantitatively compare the proposed controller performances by calculating three performance indices, IAE, ITAE, and CEF from Eqs. (2.59), (2.60), and (2.61). They assess the tracking error performance and the amount of the control effort of the system. Table 2.6 presents the performance index values of the proposed controllers for two cases.

Table 2.6: Performance measurements of the proposed control schemes.

Controller	IAE	ITAE	CEF
SDRE	1.459	7.247	0.26
SDRE Approx.	1.781	7.65	0.27
$H_2 - H_\infty$	1.021	6.147	0.72
$H_2 - H_\infty$ Approx.	1.063	6.196	0.76

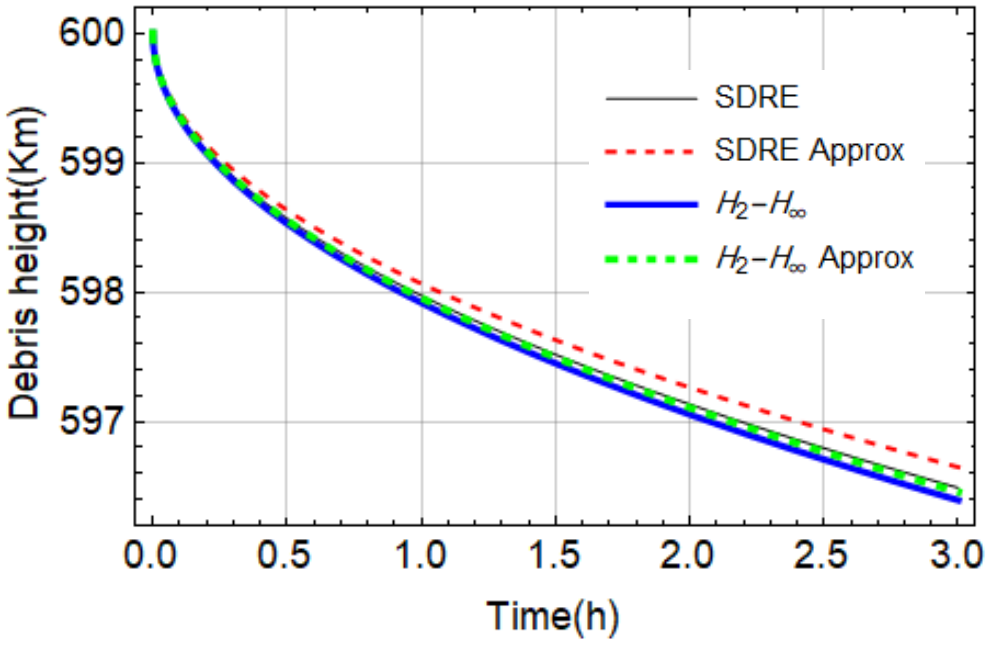


Figure 2.12: Orbital height of space debris reduced by different controllers.

The main objective is to reduce the orbital altitude of the space debris. The system brings the debris to lower orbit and reduces the orbital lifetime. The thrusters apply very low forces over a continuous time much longer than the original orbital period. This force induces a tangential ΔV on the system in the opposite direction of the original velocity. In this case, system will end up at lower but still circular orbit. Figure 2.12 depicts decreasing debris altitude. As shown in Fig. 2.11 (a), by decreasing the velocity of the satellite, there is a overshoot in the in-plane angle. The controller regulates this error and brings the system back to the azimuth direction. In addition, one can achieve more reduction in the velocity by using $H_2 - H_\infty$ controller. It happens because the controller can overcome the effect of drag disturbance on the system.

2.5.2 $H_2 - H_\infty$ Model Reference Adaptive Control of Tethered Satellite System

2.5.2.1 Problem description

This study aims to investigate the control of a triangular configuration and triple mass tethered satellite system using the robust $H_2 - H_\infty$ Model Reference Adaptive Control (HMRAC) scheme. The system is actuated by thrusters to generate control forces. The dynamical model of the semi-ideal system, which acts as a reference model is described with a known external disturbance, called the J_2 perturbation. The proposed MRAC design methodology is based on the stable semi-ideal nonlinear reference model, which is regulated by a state feedback controller using the $H_2 - H_\infty$ technique.

Then, the real system with the unknown disturbances is controlled by the feedback of the reference model control scheme. The main benefit of using the HMRAC is having robustness of the reference model, which decreases the computational burden of the classical MRAC. The numerical simulation results are presented and compared with the Linear Quadratic Regulator to demonstrate the effectiveness of the proposed control method. Also, the effectiveness of the proposed controller in improving attitude maneuverability is demonstrated.

2.6 Introduction

Adaptive control methods are widely studied for nonlinear systems to deal with planet uncertainties, time-varying plant parameters, and environment disturbances. Model Reference Adaptive Control (MRAC) is one of the powerful control schemes used for adjusting the performance of nonlinear systems. In the MRAC design, the objective is to push the response(s) of the unknown plant to track the output of the reference plant asymptotically by adjusting the controller gains recursively. The adaptive control architecture has also been studied for various classes of nonlinear systems [103–105].

In addition, optimal control methods allow trade offs between the precision and consumption of energy in the control of systems. Furthermore, the H_∞ approach have shown desirable effectiveness in the controller design in minimizing the effect of the external disturbance on the system. Its robustness criteria mixed with the optimality makes it a good candidate to use for regulating nonlinear systems. Wang *et al.* [56] presented a novel $H_2 - H_\infty$ SDRE control approach with the purpose of providing a more effective control design framework for

continuous-time nonlinear systems. In addition, the goal was to achieve a mixed nonlinear quadratic regulator and H_∞ control performance criteria.

The main objective of this chapter is to develop an adaptation mechanism for the MRAC of nonlinear systems, which is based on the adaptation of the semi-ideal model. We examine a TSS for the proposed $H_2 - H_\infty$ model reference adaptive controller.

The TSS is commonly subjected to perturbations and disturbances from space environment such as the J_2 perturbation, drag force, solar radiation pressure, heating effects, and orbital eccentricity [106]. There are different mathematical modelings for these perturbations but none of them are precise. An adaptive controller is a good candidate to regulate a TSS in the space environment. Palmerini and Sabatini [107] proposed a global frame to evaluate and select formation flying control strategies at low orbit altitudes. They considered the most significant environmental effect such as aerodynamic drag and J_2 perturbations on the system and introduced a linear quadratic regulator for the control scheme.

Here, we propose a new MRAC control design method for the TSS by extending the $H_2 - H_\infty$ SDRE method to the adaptive control [108]. We consider a stable nonlinear reference model with known disturbances. The control of the reference model is designed by using $H_2 - H_\infty$ SDRE method. Then, the control signal for the nonlinear plant is generated by using a recursive adaptation procedure such that the plant states track the states of the nonlinear reference model. In addition to its reduced computational time, the main objective here is to develop an adaptation mechanism for the MRAC of the nonlinear system, which is based on the adaptation of the $H_2 - H_\infty$ SDRE model. The proposed method allows one to design a new adaptive control algorithm for a class of uncertain nonlinear systems.

2.6.0.1 Control Design: $H_2 - H_\infty$ Model Reference Adaptive Control

The MRAC relies on a reference model, where the output of the reference model is taken as a reference to the uncertain real system. The MRAC is designed such that the difference between the output of the reference model and the real system goes to zero. The controller gains are updated recursively based on the error, reference input, and the states of the real system. The general parallel robust MRAC control scheme is depicted in Fig. 2.13.

Consider the semi-ideal reference model with a uniformly reference input $\bar{u}_m \in R^{m\times 1}$ and

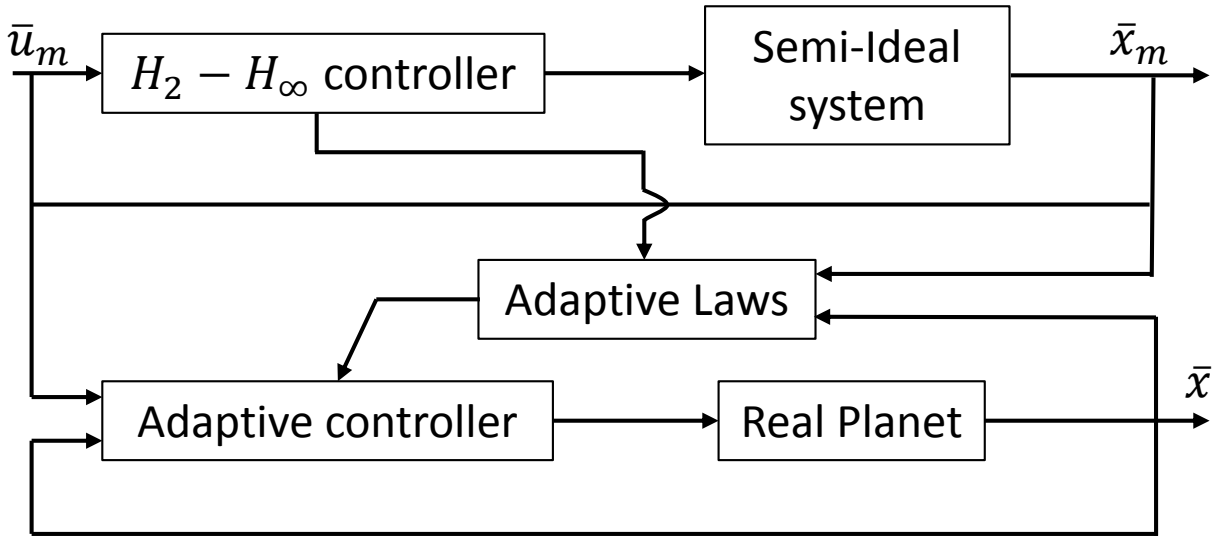


Figure 2.13: General parallel MRAC scheme.

a known disturbance is given as follows:

$$\dot{\bar{x}}_m = A_m \bar{x}_m + B_m \bar{u}_m + F_m w \tag{2.78}$$

where $\bar{x}_m \in R^{n\times 1}$ is the state vector of the reference system, $A_m \in R^{n\times n}$ is the Hurwitz, $B_m \in R^{n\times m}$, and $w \in R^{q\times 1}$ is the disturbance vector. Also, A_m, B_m, and F_m are known state dependent coefficient matrices. We should note that choice of the matrix A_m is not unique. One of the conditions is that in the domain of interest the pairs (A_m, B_m) should be pointwise controllable.

These condition could be easily verified by checking the rank of the controllability matrix. A sufficient way to test stabilizability condition is checking the controllability matrix :

$$\mathbf{C} = \left[B_m \,|A_m B_m \,|A_m^2 B_m \,|\cdots|\, A_m^{n-1} B_m\right] \tag{2.79}$$

Which has $Rank\,[\mathbf{C}] = n$. This system can be control by the $H_2 - H_\infty$ described in above. Now, we consider the adaptive part of the control structure. This part of the control structure will attempt to regulate the real system with the unknown disturbances with a similar

structure of the semi-ideal system.

In order to use the MRAC on the real system, we linearize the real system by using Tyler expansion as follows:

$$\dot{\bar{x}}(t) = A\,\bar{x} + B\Lambda\bar{u} \tag{2.80}$$

where $\bar{x} \in R^{n\times 1}$ is the states of the real system, $A \in R^{n\times n}$, $\Lambda = diag(\lambda_1 \cdots \lambda_m) \in R^{m\times m}$, and $B \in R^{n\times m}$. The controller has only knowledge about the sign of Λ and B. Furthermore, A is unknown to the controller.

To achieve the design objective of having the real system to behave as the reference model, the following matching conditions should be met. There exists constant matrices K_x and K_r such that:

$$A + B\Lambda K_x^T = A_m \tag{2.81}$$

$$B\Lambda K_r^T = B_m \tag{2.82}$$

where $K_x \in R^{n\times m}$ and $K_r \in R^{m\times m}$ are time varying control gains. Then, the proposed control law is:

$$\bar{u} = {K_x}^T\bar{x} + {K_r}^T\bar{u}_m \tag{2.83}$$

The tracking error between the semi-ideal reference system and the real planet is defined as follows:

$$\bar{e} = \bar{x} - \bar{x}_m \tag{2.84}$$

The time derivative of the tracking error is presented as follows:

$$\begin{aligned}\dot{\bar{e}} &= \dot{\bar{x}} - \dot{\bar{x}}_m \\ &= A\,\bar{x} + B\Lambda\bar{u} - (A_m\bar{x}_m + B_m\bar{u}_m + F_m w)\end{aligned} \tag{2.85}$$

by substituting Eq. (2.83) into Eq. (2.85) and adding $\pm A_m\bar{x}$, we have :

$$\dot{\bar{e}} = A_m\bar{e} - (A_m - A - B\Lambda {K_x}^T)\bar{x} - (B_m - B\Lambda {K_r}^T)\bar{u}_m - F_m w \tag{2.86}$$

We should note that the value of $F_m w$ can be neglected and other terms also dominate, so it can be eliminate from Eq. (2.86). By adding $\pm B_m K \bar{x}$, the tracking error can be rewritten as follows:

$$\dot{\bar{e}} = A_{clm}\bar{e} - A_{ad}\bar{x} - B_{ad}\bar{u}_m \tag{2.87}$$

$$A_{clm} = A_m - B_m K \tag{2.88}$$

$$A_{ad} = A_{clm} - A - B\Lambda K_x^T \tag{2.89}$$

$$B_{ad} = -B\Lambda K_r^T \tag{2.90}$$

The Lyapunov function here is selected as follows:

$$V = \bar{e}^T P_{ad}\bar{e} + \mathrm{tr}(A_{ad}P_{ad}\Gamma_x^{-1}A_{ad} + B_{ad}^T P_{ad}\Gamma_r^{-1}B_{ad}) \tag{2.91}$$

where $\Gamma_x \in R^{n\times n}$ and $\Gamma_r \in R^{n\times n}$ are diagonal constant matrices. The positive definite matrix $P \in R^{n\times n}$ is obtained by using the following Lyapunov equation:

$$P_{ad}A_{clm} + A_{clm}^T P_{ad} = -Q_{ad} \tag{2.92}$$

where Q_{ad} is a symmetric positive definite matrix. The time derivation of the Lyapunov

function is given as follows:

$$\begin{aligned}\dot{V} =&\dot{\bar{e}}^T P_{ad}\bar{e} + \bar{e}^T P_{ad}\dot{\bar{e}} \\ &+ \mathrm{tr}\left(\frac{d}{dt}(A_{ad}P_{ad}\Gamma_x^{-1}A_{ad} + B_{ad}^T P_{ad}\Gamma_r^{-1}B_{ad})\right) \\ =&(A_{clm}\bar{e} - A_{ad}\bar{x} - B_{ad}\bar{u}_m)^T P_{ad}\bar{e} + \bar{e}^T P_{ad}(A_{clm}\bar{e} - A_{ad}\bar{x} - B_{ad}\bar{u}_m) \\ &+ \mathrm{tr}\left(\frac{d}{dt}(A_{ad}P_{ad}\Gamma_x^{-1}A_{ad} + B_{ad}^T P_{ad}\Gamma_r^{-1}B_{ad})\right) \\ =&\bar{e}^T(P_{ad}A_{clm} + A_{clm}^T P_{ad})\bar{e} \\ &- \bar{x}^T A_{ad}P_{ad}\bar{e} - \bar{e}^T P_{ad}A_{ad}\bar{x} \\ &- \bar{u}_m^T B_{ad}P_{ad}\bar{e} - \bar{e}^T P_{ad}B_{ad}\bar{u}_m \\ &+ \mathrm{tr}\left(\frac{d}{dt}(A_{ad}P_{ad}\Gamma_x^{-1}A_{ad} + B_{ad}^T P_{ad}\Gamma_r^{-1}B_{ad})\right) \end{aligned} \tag{2.93}$$

with regards of the properties of trace of the vectors, Eq. (2.93) can be rewritten as follows:

$$\begin{aligned}\dot{V} =& -\bar{e}^T Q_{ad}\bar{e} + 2\mathrm{tr}(A_{ad}P_{ad}\Gamma_x^{-1}\dot{A}_{ad} - A_{ad}P_{ad}\bar{e}\bar{x}^T) \\ &+ 2\mathrm{tr}(B_{ad}^T P_{ad}\Gamma_r^{-1}\dot{B}_{ad} - B_{ad}P_{ad}\bar{e}\bar{u}_m^T)\end{aligned} \tag{2.94}$$

To have $\dot{V} < 0$, the following relations should be met:

$$\dot{A}_{ad} \leq \Gamma_x \bar{e}\bar{x}^T \tag{2.95}$$

$$\dot{B}_{ad} \leq \Gamma_r \bar{e}\bar{u}_m^T \tag{2.96}$$

Time derivative of two matrices are as follow:

$$\dot{A}_{ad} = \dot{A}_{clm} - \frac{d}{dt}|A| - \frac{d}{dt}|B|\Lambda K_x^T - |B|\Lambda\dot{K}_x^T \tag{2.97}$$

$$\dot{B}_{ad} = -\frac{d}{dt}|B|\Lambda K_r^T - |B|\Lambda\dot{K}_r^T \tag{2.98}$$

If we assume the boundary for actual system as $|A| = A_{max}$ and $|B| = B_{max}$, and the time derivative of constant matrix bound is zero, so Eqs. (2.95) and (2.96) can be written as

follow:

$$-\left((B_{max}^T B_{max})^{-1}\Lambda B_{max}^T\right)\Gamma_x \bar{e}\bar{x}^T \leq \dot{K}_x^T \tag{2.99}$$

$$-\left((B_{max}^T B_{max})^{-1}\Lambda B_{max}^T\right)\Gamma_r \bar{e}\bar{u}_m^T \leq \dot{K}_r^T \tag{2.100}$$

These differential equations define the adaptive gian of control law. It is better to change the inequality to following equations to demonstrate the adaptation laws as follow:

$$\dot{K}_x = -\Gamma_x\, \bar{x}\, \bar{e}^T \left((B_{max}^T B_{max})^{-1}\Lambda B_{max}\right) - \sigma_x G_x \tag{2.101}$$

$$\dot{K}_r = -\Gamma_r\, \bar{u}_m\, \bar{e}^T \left((B_{max}^T B_{max})^{-1}\Lambda B_{max}\right) - \sigma_r G_r \tag{2.102}$$

where σ_x and σ_r are positive constants, which they are playing an effective role in the control design, and G_x and G_r matrices are formulated as follows:

$$G_x = K_x\left(1 - \frac{\|K_x\|}{K_x^{max}}\right) \tag{2.103}$$

$$G_r = K_r\left(1 - \frac{\|K_r\|}{K_r^{max}}\right) \tag{2.104}$$

which, $\|.\|$ is the norm of the vector.

2.6.0.2 Tethered Satellite System Modeling

The dynamics governing the TSS are highly nonlinear, which include translational motion of the entire system around the Earth, relative attitude motion of the configuration, and the length of the tether. We consider a triple mass TSS with three long tethers connected to each two satellites at their ends. The tether is assumed to be a straight, inextensible, and massive bar. This assumption let the system have two degree of freedom. The angles θ and ϕ are the in-plane and out-of-plane libration ones taken as the states of the system. The elasticity of the tether adds the additional nonlinearity to the dynamics of the system. Here, in this system, one can consider the elasticity of tethers as an uncertainties, which we do not take into account. Furthermore, the tethered satellites are considered as lumped masses due to the large ratio of tether length over the satellites' dimensions.

The proposed system consists of three equal masses, m which are connected with tethers

of length l and mass m_t. The triple mass TSS configuration is shown in Fig. 2.14. It should be noted that r is the center of mass G position vector relative to the center of the Earth and r_1, r_2, and r_3 are the satellite relative positions to the center of mass. Due to the complexity of the system dynamics, the equations of motion are derived using the Lagrangian formulation as follows [100]:

$$\begin{aligned} &\mathbf{L} = T_{total} - V_{total}; \\ &\frac{\partial}{\partial t}[\frac{\partial \mathbf{L}}{\partial \dot{q}_i}] - \frac{\partial \mathbf{L}}{\partial q_i} = Q_i \end{aligned} \tag{2.105}$$

where $\mathbf{L}$, T_{total}, and V_{total} are the Lagrangian function, total kinetic energy, and total potential energy, respectively. In addition, Q_i and q_i are the external torques for the ith state of the system and the independent states, respectively.

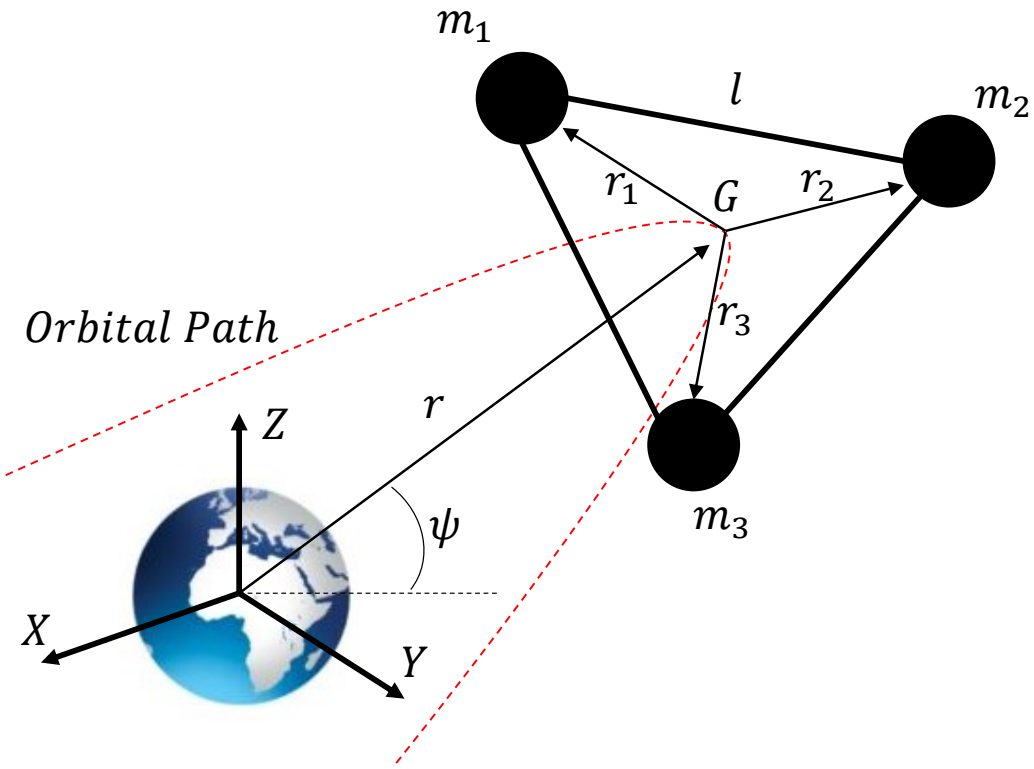


Figure 2.14: TSS configuration.

Summation of the kinetic energies for the end-masses and the integrated kinetic energy across the length of the tether is the total kinetic energy of the system:

$$\begin{aligned} T_{total} &= \sum_{i=1}^{3} m_i + \sum_{i=1}^{3} m_{t_i} \\ &= \frac{3}{2}(m+m_t)\dot{r}^2 + \frac{l^2}{4}m\left(\dot{\phi}^2 + (\dot{\theta}+\omega_0)^2\right) \\ &\quad + \frac{l^2}{4}(m+m_t)\left((\dot{\theta}+\omega_0)\cos\phi\right)^2 \end{aligned} \tag{2.106}$$

The total potential energy is given by:

$$\begin{aligned} V_{total} &= \sum_{i=1}^{3} V_{m_i} + \sum_{i=1}^{3} V_{m_{t_i}} \\ &= \frac{\mu_g(U_0+U_{J_2})}{r}(3(m+m_t) \\ &\quad + \frac{m}{4}\left(\frac{l}{r}\right)^2(1-3\sin^2\phi\cos^2\theta)) \\ U_0 &= -1 \;; \quad U_{J_2} = \left(\frac{R_E}{r}\right)^2\frac{J_2}{2}(3\sin^2\psi - 1) \end{aligned} \tag{2.107}$$

where $\mu_g = 3.986\times10^5\ Km^3/s^2$ is the gravitational parameter of the Earth, $J_2 = 1.082\times10^{-3}$ is the harmonic coefficient, ψ is the latitude of the r, and R_E is the radius of the Earth.

The equations of motion of the system without any environmental disturbances are given as:

$$\begin{aligned} &\frac{l^2}{2}(\ddot{\theta}+\dot{\omega}_0)\left[m+(m+m_t)\cos^2\phi\right] - \\ &\quad l^2(\dot{\theta}+\omega)(m+m_t)\dot{\phi}\sin\phi\cos\phi - G_\theta = u_\theta \end{aligned} \tag{2.108}$$

$$\frac{l^2}{2}m\ddot{\phi} + \frac{l^2}{2}(\ddot{\theta}+\dot{\omega}_0)^2(m+m_t)\sin\phi\cos\phi - G_\phi = u_\phi \tag{2.109}$$

where ω_0 is the orbital mean motion, G_θ and G_ϕ are the corresponding gravitational forces for each state, u_θ and u_ϕ are the control inputs. The first disturbance acting on the TSS is the J_2 perturbation. This perturbation has been completely modeled before [102], so we consider it as a known perturbation. By neglecting the higher order terms of the Earth's

spherical harmonics, the gravitational terms are given as follow:

$$G_\theta = -\frac{3\,m\,\mu_g}{2r}\left(\frac{l}{r}\right)^2 (U_0 + U_{J_2}) \sin\theta\cos\theta\sin^2\phi \tag{2.110}$$

$$G_\phi = -\frac{3\,m\,\mu_g}{2r}\left(\frac{l}{r}\right)^2 (U_0 + U_{J_2}) \sin\phi\cos\phi\cos^2\theta \tag{2.111}$$

There are some perturbations on the TSS subject to space environment effects such as the drag force, the solar pressure, the heating effect, and the orbital eccentricity. These perturbations have been investigated before [100, 106, 109, 110], but they are not fully modeled. We consider them as unknown disturbances to the real system.

2.6.0.3 Controllability and Accessibility

In this section, we use the Lie-brackets theory to investigate the accessibility of the system. The mathematical framework for this analysis is based on the method presented in [111]. This method is briefly explained here. A control affine nonlinear system can be expressed as:

$$\dot{q} = f(q) + \sum_{i=1}^{m} g_i(q)\,u_i \tag{2.112}$$

where $q \in R^n$ are state variables, f is the drift vector field, g_i is the i^{th} input vector field and u_i is the control input. For this nonlinear system, starting from an initial state q_0 there exists a control that can drive the system to a space $H(q_0)$, which is formed by system vector fields and their nested Lie-brackets given as:

$$H = span\left(G^0, G^1, G^2, \cdots, G^m\right) \tag{2.113}$$

where

$$
\begin{aligned}
&G^0 \equiv \{g_1\} \\
&G^1 \equiv \{[g_1, g_2]\} \\
&G^2 \equiv \{[g_1, [g_2, g_3]]\} \\
&\vdots \\
&G^m \equiv \{[g_1, [g_2, [\cdots, g_m]]]\} \\
&i = 1, 2, \cdots, m;
\end{aligned} \tag{2.114}
$$

where the Lie-bracket of vector fields is defined as:

$$
[X, Y] = L_X Y = Y\frac{\partial X}{\partial x} - X\frac{\partial Y}{\partial x} \tag{2.115}
$$

The nonlinear system is fully Accessible if $Rank(span(L_o)) = n$, where $L_o \equiv [f, [G_0, [\cdots, G_m]]]$ [30]. We verify the accessibility of the system by using Eq. (3.15). For this purpose, Eqs. (2.108) and (2.109) can be written in affine-control form as Eq. (3.13). The rank of L_o is found as 4 so the system is controllable.

2.6.0.4 Stability Analysis

In this section, we analyze the stability of the system in neighborhoods of equilibria. In this regard, we use *Poincare Map* and *Floquet Theory*. Since closed form analytic solution of differential equations of motion is not available, numerical integration is used [47].

Let $q(t)$ be the the periodic solution of the equations of motion for all states. The simplified hyper plane of the Poincare section can be defined as:

$$
\mathbf{S} = (\theta, \dot{\theta}, \phi, \dot{\phi}) \in R^4 \tag{2.116}
$$

If $q[k] \in \mathbf{S}$ denotes the k^{th} intersection of $\mathbf{S}$ by the flow of $q(t)$ the discrete-time Poincare Map $\mathbf{P} : \mathbf{S} \to \mathbf{S}$ can be expressed as:

$$
q[k+1] = \mathbf{P}(q[k]) \tag{2.117}
$$

Subsequently, if q^* stands for fixed point of the Poincare Map, then the local exponential stability of q^* on $\mathbf{S}$ is equivalent to local exponential stability of the system. In this regard, the local linearizion of the Poincare Map about its fixed point yields

$$\begin{aligned} \mathbf{P}(q) &\simeq \mathbf{J}(q^*)(q - q^*) \\ \mathbf{J} &= \frac{\partial \mathbf{P}(q)}{\partial q} \end{aligned} \tag{2.118}$$

where $\mathbf{J}$ is 4×4 linearized Jacobian matrix of $\mathbf{P}$. Then, the floquet or characteristic multipliers ρ_i's are defined as eigenvalues of the Jacobian matrix $\mathbf{J}$

$$\rho_i = \|\lambda_i\| \, ; \lambda_i = \text{eig}(\mathbf{J}) \tag{2.119}$$

where ρ_i and λ_i are the i^{th} characteristic multiplier and Jacobian matrix eigenvalue, respectively, and "$\| \, \|$" is the modulus operator. Therefore, the stability can be identified by Floquet multiplier metrics as:

$$q(t) = \begin{cases} \text{stable} & \forall \, \|\rho_i\| < 1 \\ \text{marginally stable} & \forall \, \|\rho_i\| = 1 \\ \text{unstable} & \forall \, \|\rho_i\| > 1 \end{cases} \tag{2.120}$$

We conduct numerical simulation examples to observe the stability of the semi-ideal system. Typical tether parameters and the orbit conditions are listed in Table 3.1, which are used for the numerical simulation. A sample sun-synchronous circular orbit of 600 km is selected for the analysis of the semi-ideal reference system.

Table 2.7: The detail of TSS parameters and the initial orbital elements in the numerical simulation.

parameter	value	parameter	value
m	50 Kg	ω	90°
m_t	10 Kg	e	0.01
l	1 Km	i	97.8°
ω_0	0.001083 rad/s	Ω	270°

The Floquet multipliers associated with the corresponding Poincare map for the simulation of fixed point $\{q_1^* : [\theta = 0°; \dot{\theta} = 0\,\mathrm{deg}/s; \phi = 0°; \dot{\phi} = 0\,\mathrm{deg}/s]\}$ are calculated numerically as:

$$\begin{aligned}
\rho_1 &= \|\lambda_1\| = 0.0025 < 1 \\
\rho_2 &= \|\lambda_2\| = 0.0012 < 1 \\
\rho_3 &= \|\lambda_3\| = 0.3866 < 1 \\
\rho_4 &= \|\lambda_4\| = 0.00374 < 1
\end{aligned} \tag{2.121}$$

All norms of characteristic multiplier of the semi-ideal system are less than one, which means that the semi-ideal system is asymptotically stable. We choose another fixed point as: $\{q_2^* : [\theta = 270°; \dot{\theta} = 0\,\mathrm{deg}/s; \phi = 90°; \dot{\phi} = 0\,\mathrm{deg}/s]\}$. The Floquet multipliers associated with the corresponding Poincare map for this simulation are also calculated numerically as:

$$\begin{aligned}
\rho_1 &= \|\lambda_1\| = 0.0106 < 1 \\
\rho_2 &= \|\lambda_2\| = 0.0058 < 1 \\
\rho_3 &= \|\lambda_3\| = 2.8745 > 1 \\
\rho_4 &= \|\lambda_4\| = 0.875 < 1
\end{aligned} \tag{2.122}$$

As one can see, one of the norms is greater than one, which means that the semi-ideal system is unstable at that fixed point.

2.6.0.5 Numerical Simulation

This section presents numerical simulations of two scenarios to illustrate the performance of the proposed control method. For this purpose, we simulate two different scenarios including a stable condition reaching and a rest-to-unstable attitude maneuver. The physical characteristics of the system and the orbital elements are chosen according to Table 2.7.

The following design parameters are chosen for the semi-ideal system as follow:

$$F_m = [0.1]\,;\;\; Q = 100I_4\,;\;\; R = 0.1I_2\,;\;\; \gamma = 0.2 \tag{2.123}$$

The selected initial conditions for all scenarios are listed in Table 2.8.

Table 2.8: The initial conditions fo the numerical simulation.

	θ	$\dot{\theta}$	ϕ	$\dot{\phi}$
Scenario 1	20°	$+1\,\mathrm{deg}/s$	10°	$-1\,\mathrm{deg}/s$
Scenario 2	0°	$0\,\mathrm{deg}/s$	0°	$0\,\mathrm{deg}/s$

2.6.1 Scenario 1: Reaching stable point

Now, we present the example for reaching stable condition from an arbitrary initial condition. The first scenario is implemented to illustrate the stability and tracking performance of the HMRAC under unknown perturbation effects. Also, we use a white noise as an unknown disturbance to the system. The tracking performance improvement by the HMRAC is verified by comparison with the LQR. For this scenario, a comparison between two controllers ,$H_2 - H_\infty$ controller on the semi-ideal system and MRAC on the uncertain real system, converge to the desired values as $\{\theta = 0^\circ; \dot{\theta} = 0\,\mathrm{deg}/s; \phi = 0^\circ; \dot{\phi} = 0\,\mathrm{deg}/s\}$. Obviously, both the LQR and the proposed HMRAC can achieve zero steady position tracking errors.

For our proposed robust adaptive control design, the baseline control is introduced to accomplish required transient and steady control performances for the formation model under unknown external disturbances. The HMRAC is employed to make the system robust to the uncertainty effects and external disturbances and recover the control performance defined by the baseline controller as much as possible. The tracking performance is significantly improved using much fewer control efforts as illustrated in Figs. 2.15-2.16.

The control efforts can be quantified as the root mean square of the deviations of the control inputs. The quantification results are shown in Table 2.9. The proposed controller is much more efficient than the LQR control scheme.

Table 2.9: Root mean square (rms) of control efforts at Scenario 1.

	$\text{rms}_{u\theta}$	$\text{rms}_{u\phi}$
$H_2 - H_\infty$	2.48	1.87
HMRAC	2.36	1.58
LQR	4.19	5.36

2.6.2 Scenario 2: Rest-to-unstable maneuver

This scenario is adopted to test the performance of the proposed controller in order to change the attitude of the TSS from a stable point (rest point) to unstable condition. The same orbital and control parameters of the $H_2 - H_\infty$ used in the first scenario are utilized here. The unstable condition is chosen to be as $\{\theta = 270°;\ \dot{\theta} = 0\,\text{deg}/s;\ \phi = 90°;\ \dot{\phi} = 0\,\text{deg}/s\}$.

The position tracking performances of the LQR and the proposed robust adaptive design are shown in Figs. 2.17-2.18. As one can see, for this scenario, both of the two controllers $H_2 - H_\infty$ and HMRAC have similar performances. They both regulate the semi-ideal and real systems to the unstable point. The tracking errors of the proposed design have much smaller root mean squares than the LQR as shown in Table 2.10, which also illustrates the effectiveness of the proposed design.

Table 2.10: Root mean square (rms) of tracking stated and control efforts at Scenario 2.

	rms_{θ}	rms_{ϕ}	$\text{rms}_{u\theta}$	$\text{rms}_{u\phi}$
$H_2 - H_\infty$	3.15	1.61	3.48	4.68
HMRAC	3.24	1.67	3.71	4.89
LQR	1.92	6.56	4.11	5.03

Figure 2.17 shows the time histories of the in-plane and out-of-plane liberation angles of the TSS. As one can see, the attitude response of the system is properly controlled to unstable position and the proposed control scheme has less percents overshoot of the response of the real system than the LQR.

In general, real time implementation of the proposed control method is computationally taxing. This is particularity true for high order systems. In future work, we will propose another controller to overcome with this drawback.

2.7 Conclusions

A novel optimal robust nonlinear controller was investigated in this study. This control scheme was based on the SDRE approach. We showed that the generalized Riccati equation can be cast in the traditional form and solved with standard solvers. Then, an approximation method was presented, which was based on least squares technique and an offline neural network weight updated scheme. The objective of this approximation method was to facilitate real time implementation of the proposed methods. We conducted different simulation runs on different applications to validate our proposed controller.

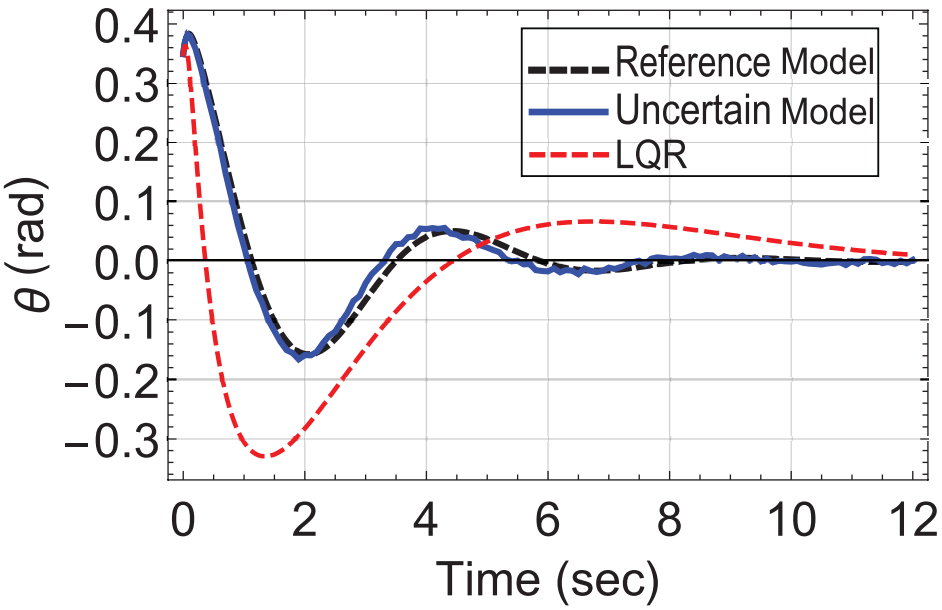
The first application was a flywheel-based inverted pendulum system. We numerically compared the performances of the conventional SDRE, approximation approach, and $H_2 - H_\infty$ controllers with the proposed RNQG controller. We presented comparison of controller outcomes for three different cases. All controllers were applied to an ideal system to show the validity of the approximation method. Then, the response of the idealized system subject to disturbance and two noise levels was obtained using all controllers. In simulation, all controllers were able to stabilize the system. However, the performance of the RNQG was superior to the others. The proposed controller outperformed the others because of its capability in dealing with disturbance and process noise.

Also, we investigated the effect of three parameters, time delay, sensor noise, and parameter uncertainty on the outcomes of SDRE and RNQG control schemes. The control action was considered to be successful when a solution to the Riccati equation can be found. Using this criteria of success, we identified the regions of parameter space that resulted in effective control action. Subsequently, we found the enclosures of effective control regions in the parameter space for two control methods: SDRE and RNQG. We showed that the success volume of the proposed RNQG controller is significantly larger than the success volume of the SDRE controller.

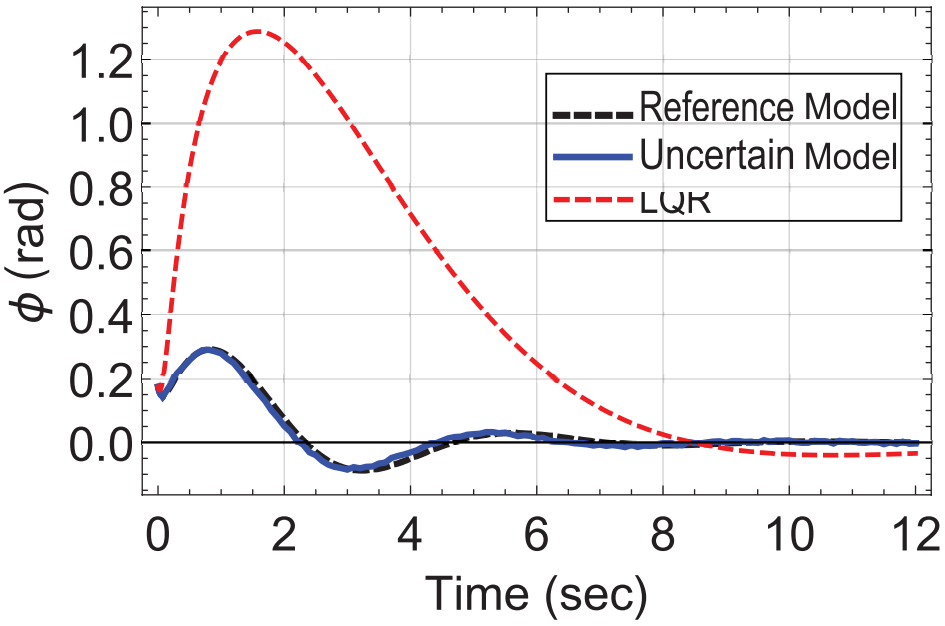
Secondly, the performance of the $H_2 - H_\infty$ controller to detumble a Tethered-Satellite-

Debris (TSD) system has been investigated. The controller showed the ability to reject the effect of external disturbances such as the aerodynamic drag acting on the nonlinear system. The results depicted convergence of the libration angles and regulation of the large errors of in-plane angle caused by the decrease in the system velocities. Reduced velocities forced the system to descend to lower orbits, which led to accelerated debris removal.

At the end, a new robust $H_2 - H_\infty$ model reference adaptive control of a triple mass tethered satellite system in the presence of external disturbances was studied. The dynamical model of the semi-ideal system was formulated through a known disturbance called the J_2 perturbation. The semi-ideal system , which acts as a reference model was controlled by the $H_2 - H_\infty$ controller. An adaptive controller was utilized to regulate the uncertain real model of the TSS under space environmental unknown disturbances. An appropriate control performance was achieved using $H_2 - H_\infty$ model reference adaptive control method. The introduced control scheme was a useful approach to reduce the uncertainty and disturbance effects on the complex real system. Numerical simulation results demonstrated the effectiveness of the proposed approach in achieving desirable performance and maneuverability. Since the controller is robust and adaptive simultaneously, it will be effective under the system's uncertainties, such as elasticity of the tethers and the additional nonlinearity added to the system.

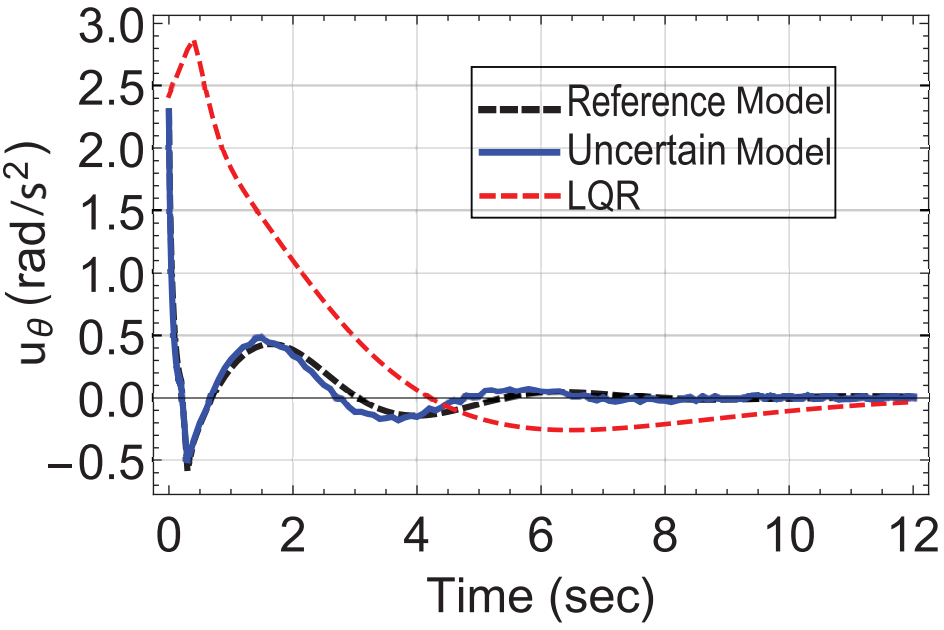


(a) In-plane libration angle

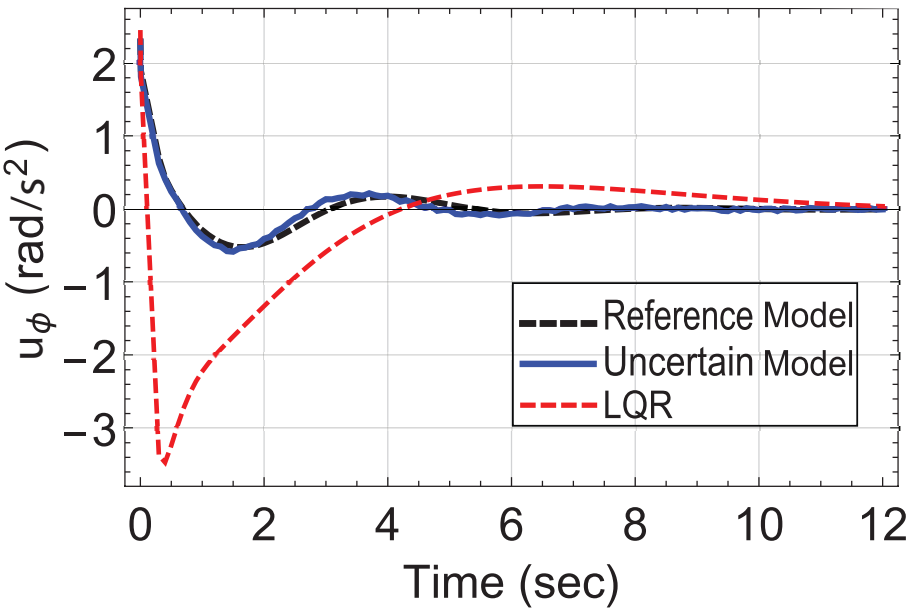


(b) Out-of-plane libration angle

Figure 2.15: Comparison of the time histories of the system's states for scenario 2.

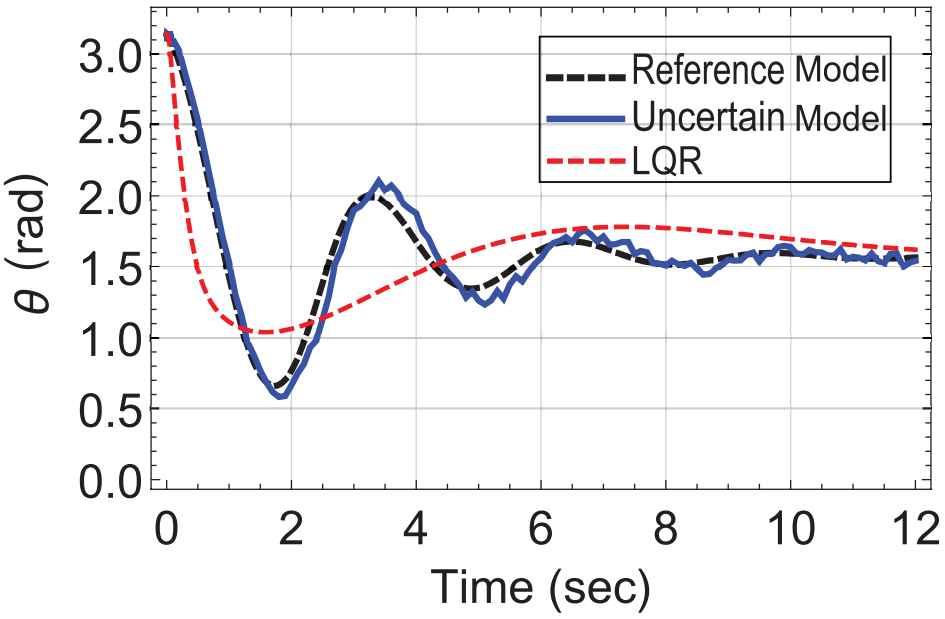


(a) Control input for θ

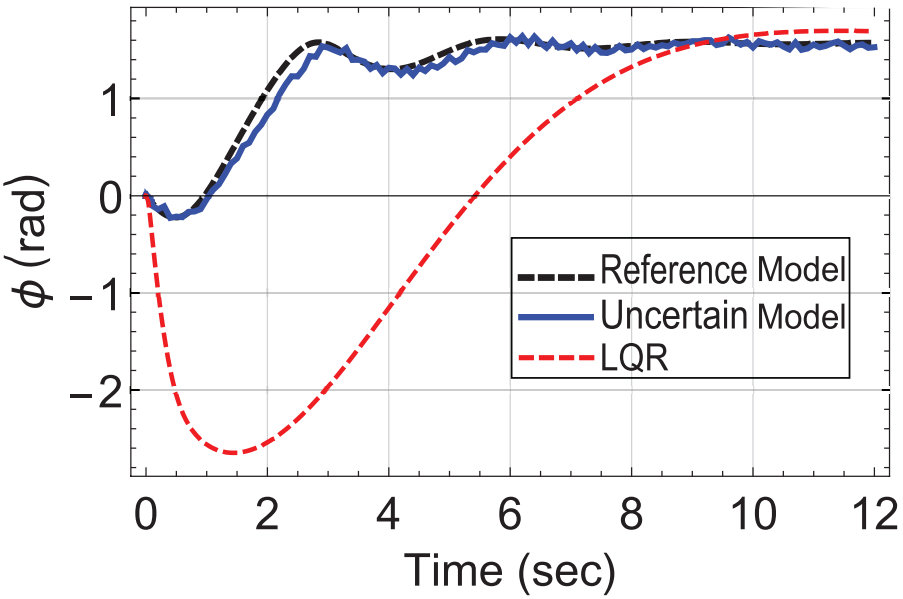


(b) Control input for ϕ

Figure 2.16: Simulation results of the control inputs for scenario 1.

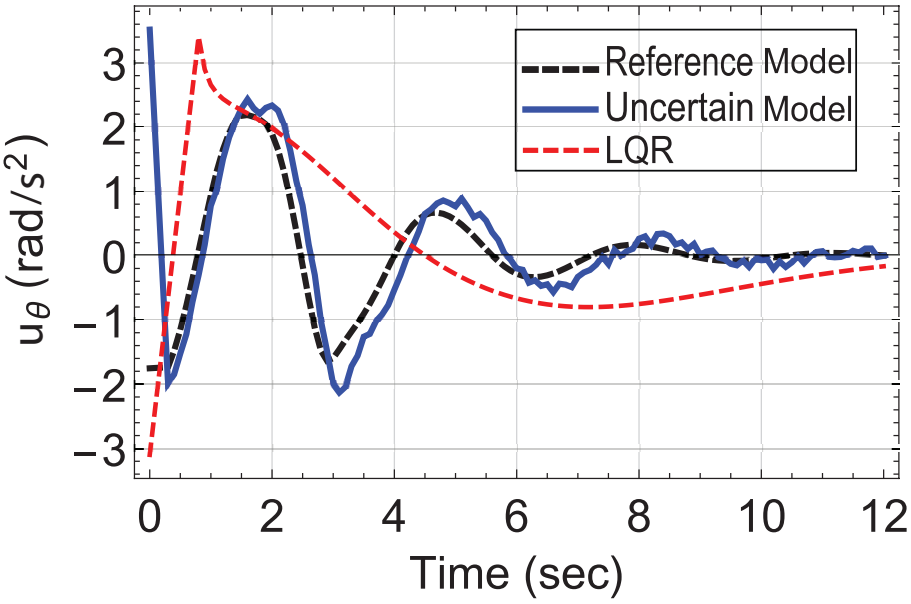


(a) In-plane libration angle

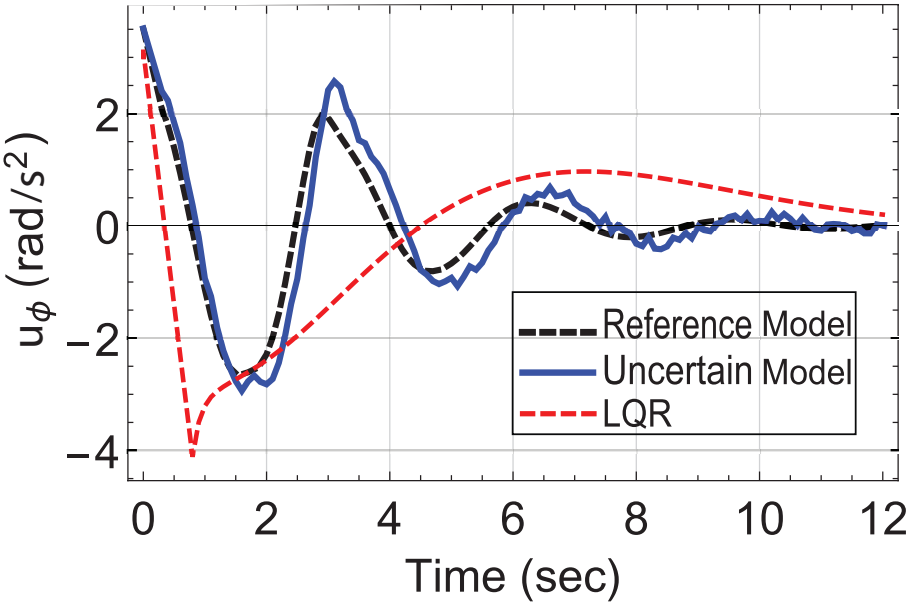


(b) Out-of-plane libration angle

Figure 2.17: Comparison of the time histories of the system's states for scenario 2.



(a) Control input for θ



(b) Control input for ϕ

Figure 2.18: Simulation results of the control inputs for scenario 2.

Chapter 3

Nonlinear Dynamics and Control of An Inertially Actuated Jumper Robot

3.1 Problem description

We introduce a unique hopping robot based on the inertial actuation concept that can navigate in 3D environment. Inertial actuators are used to drive the robot, where rotating mass spinners provide the jumping force while flywheels stabilized and controlled the orientation of the robot. The robot has two modes of motion, flying and ground contact modes. A mathematical model is developed and equations of motion for both modes are obtained using both Lagrangian method and Euler's moment equations. An adaptation control scheme is developed in order to generate periodic inertial actuation jumping in contact mode. Tracking controllers are also expanded to stabilize the robot in two modes and control the orientation in flying mode.

Two tracking controllers are addressed in this chapter. First, Sliding Mode Control (SMC) is designed based on Lyapunov approach. Second, State-Dependent Riccati Equation (SDRE) is also developed. Then, the nonlinear stability is analyzed. We demonstrate that successful control actions that drives the system toward a stable periodic orbit is possible. Finally, the simulation results validate the proposed controllers and the comparison between these two controllers are presented.

3.2 Introduction

The study of jumping robots is one of the active robotic research areas in recent years. Many robots with jumping ability have been built [112]. Most of them are inspired by animal behavior such as frogs and insects. Performance of jumping motion depends on how efficiently the jumping force generates. There have been different methods to generate the jumping force.

Burdick and Fiorini [113] designed miniaturized jumping robots that are based on stor-

age spring energy. Lambrecht *et al.* [114] worked on a robot that slowly retracts a jumping mechanism and it is suddenly released to generate a powerful jump. Dubowsky *et al.* [115] developed a robot using dielectric elastomer actuators to generate autonomous hops. Chan and Liu [116] designed a small-sized quadruped robot using compliant mechanisms to accomplish walking, turning, and jumping motions. Spröwitz *et al.* [117] designed a novel compliant quadruped robot, called "Cheetah-cub". The leg configuration is based on a spring-loaded, pantograph mechanism with multiple segments. Tsukagoshi *et al.* [118] created a high jumping rescue robot using a pneumatic cylinder. Bartlett *et al.* [119] described a soft robot powered by the combustion of butane and oxygen to perform untethered jumping. Broglito and his colleagues [120, 121] proposed a mechanical systems named "Juggler" those can hop and runs.

Zoghzoghy *et al.* [28] produced a baton locomotor capable of generating tapping gaits as a part of a family tree of a research study, which has been done in our lab, the System's Lab [122–127]. They used double-action inertial actuation scheme to drive the system. In addition to these robots, one can find a family tree of robotic systems presented by Tavakoli and Hurmuzlu [126, 128] and Kashki and Hurmuzlu [29, 129] that used the inertial actuation to generate jumping motions.

We intend to develop a nonlinear control scheme, which is robust to parameter variations and rejects the disturbances. Sliding Mode Control is a robust nonlinear controller that can be used to control hopping robots. Many nonlinear control approaches have been proposed to control different applications. These include optimal control, fuzzy control, adaptive control [130], robust control [131], feedback linearization [132], and sliding mode control. These methods used in different applications such as trajectory tracking in robotics [133,134].

Cui *et al.* [135] developed a novel integral SMC for underwater robotics. Their controller provided an asymptotic convergence of both tracking and estimation errors in the presence of unknown external disturbances and model uncertainties. We use SMC to stabilize and control trajectory tracking of jumping robot.

In addition, optimal control methods allow trade-offs between the precision and consumption of energy in control of robotic systems. In this chapter, we use second control methodology that is based on State-Dependent Riccati Equation (SDRE). This method

can be considered as a nonlinear version of the linear quadratic regulator (LQR) optimal method. This control method has the advantage of using the nonlinear system model directly in the LQR-like optimal control design. The critical design issue is how to establish the state-dependent coefficient (SDC) matrix. This nonlinear optimal control has been recently applied to practical systems such as robotics [136, 137].

Also, we use an adaptation controller presented in [129] to control continuous jumping motion of the robot. Kashki *et al.* designed a jumping robot that has only vertical motion. We add in xy-plane motion to that robot to establish trajectory tracking motion with continuous jumps.

Compared to other similar robots [138, 139], the proposed hopping robot can theoretically jump higher by using inertial actuation system. In addition, the robot has separate mechanism to stabilize and control the orientation while it is jumping. The combination of this control mechanism and the inertial actuation on the same robot is the originality of our presented robot. The inertial actuation system acted as the jumping mechanism can be placed anywhere in the body. However, controlling the jumping motion and body orientation are separate from each other, they have a connection in contact mode of the system. We control the attitude of the body in the air to achieve desired contact orientation to track the desired trajectory. Although several nonlinear control scheme has been used for the jumping robot, it is first time that SDRE controller is used in a jumping robot.

3.3 Dynamical model

3.3.1 System Description

Here, we briefly describe the jumping robot. An inertially actuated system makes continuous jumps. Two mass spinners are attached to the center of the body with revolute joints to generate the inertial actuation. Also, two springs are connected to the bottom of the robot in order to store the elastic energy. In addition, the robot is equipped with three reactions wheels to manipulate the orientation of the robot when it is in the air (Fig. 3.1).

The motion of the hopping robot can be divided into two modes, "Contact Mode" (CM) and "Flying Mode" (FM). The first mode is where the robot is in contact with the ground. Meanwhile, the second mode is where the robot is in the air. During these two phases, the

spinners compress the springs to generate the jumping force in CM. Then, the flywheels can be used to change the orientation of the robot during FM. Changing the orientation of the robot will indirectly enable the controllers to achieve the desired motion. Next, we will describe the physical parameters of the hopper.

The total mass of the hopper, M_T includes the main robot, spinners, wheels, and springs masses. The parameters L and L_{sp} are the natural un-stretched length of the spring and the length of the spinner's arm, respectively. The parameters k_s and b_s are the spring stiffness and damping coefficients respectively. Figure 3.1 depicts the global reference frame XYZ, and the body frame $x_b y_b z_b$, centered at the center of mass of the system.

The state variables of the system are $\{x,\, y,\, z,\, \theta,\, \phi,\, \psi,\, \alpha\}$, where $\{x,\, y,\, z\}$ are the coordinates of the center of mass, $\{\theta,\, \phi,\, \psi\}$ are the body frame angles, and α is the spinners angle relative to the z_b-axis in the counter-clockwise direction.

3.3.2 Equations of the Motion

The equations of motion are derived using Lagrange's method (3.1) and Euler's moment equation (3.2). As we mentioned before the robot has two distinct modes of motion. Because of this behavior, the hopping robot can be considered as a hybrid system.

The first set of equations of motion can be obtained by substituting the total kinetic (T_{total}) and potential (V_{total}) energies of the system into the Lagrangian ($\mathfrak{L}$) equation.

$$\begin{aligned} \mathfrak{L} &= T_{total} - V_{total} \\ \frac{\partial}{\partial t}[\frac{\partial \mathfrak{L}}{\partial \dot{q}_i}] - \frac{\partial \mathfrak{L}}{\partial q_i} &= Q_i \end{aligned} \tag{3.1}$$

where q_i and Q_i are the i^{th} generalized coordinate and generalized force, respectively. The orientation equations of the robot are found by using the Euler's moment equation.

$$\dot{H}(t) + \Omega(t) \times H(t) = M(t) \tag{3.2}$$

where $H(t)$ is the angular momentum of the robot, and $M(t)$ is external moment acted on

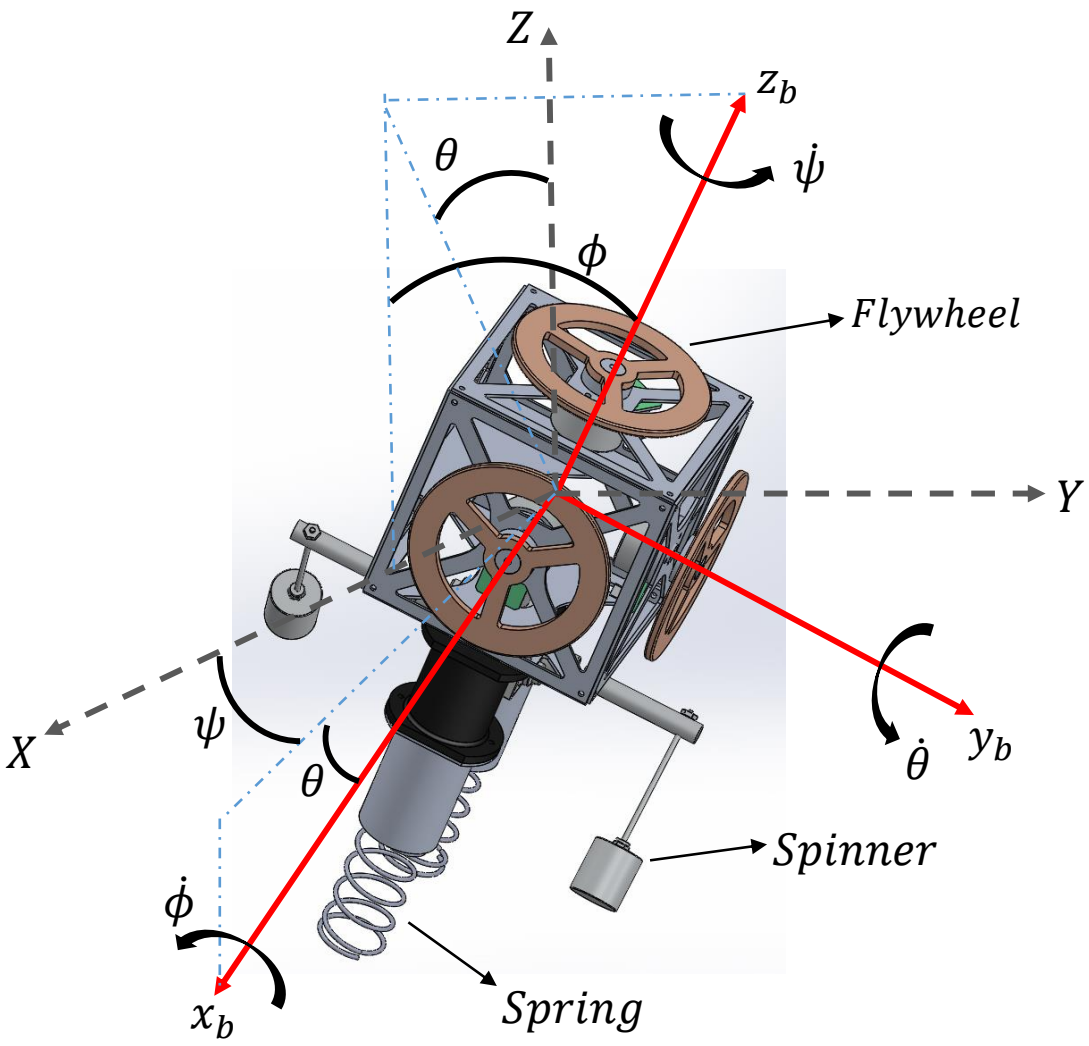


Figure 3.1: Schematics of the robot with the reference frames.

the system. $\Omega(t)$ is the angular velocity vector defined as:

$$\Omega(t) = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T \tag{3.3}$$

The equations of the motion of the system are defined as:

$$M_T \ddot{x} + S\left(z, \dot{z}\right)\left(\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi\right) = 0 \tag{3.4}$$

$$M_T \ddot{y} + S\left(z, \dot{z}\right)\left(\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi\right) = 0 \tag{3.5}$$

$$M_T \ddot{z} + 2\, L_{sp}\, m\, \left(\dot{\alpha}^2\, \cos\alpha + \ddot{\alpha}\, \sin\alpha\right) + M_T\, g - S\left(z, \dot{z}\right) \cos\theta \cos\phi = 0 \tag{3.6}$$

$$2\, L_{sp}\, m\left(g\, \sin\alpha + \sin\alpha\, \ddot{z}\, + L_{sp}\, \ddot{\alpha}\right) = \tau_s \tag{3.7}$$

$$I_{xx}\ddot{\phi} - I_{xy}\ddot{\theta} - I_{xz}\ddot{\psi} + (I_{yy} - I_{zz})\dot{\theta}\,\dot{\psi} + I_{yz}(\dot{\psi}^2 - \dot{\theta}^2) + I_{xy}\dot{\phi}\,\dot{\psi} - I_{xz}\dot{\phi}\,\dot{\theta} + \nu\, M_T\, g\, z \tan\phi = \tau_\phi \tag{3.8}$$

$$I_{yy}\ddot{\theta} - I_{xy}\ddot{\phi} - I_{yz}\ddot{\psi} + (I_{xx} - I_{zz})\dot{\psi}\,\dot{\phi} + I_{xz}(\dot{\phi}^2 - \dot{\psi}^2) + I_{yz}\dot{\phi}\,\dot{\theta} - I_{xy}\dot{\theta}\,\dot{\psi} - \nu\, \frac{M_T\, g\, z}{\cos\phi} \tan\theta = \tau_\theta \tag{3.9}$$

$$I_{zz}\ddot{\psi} - I_{xz}\ddot{\phi} - I_{yz}\ddot{\theta} + (I_{yy} - I_{xx})\dot{\theta}\,\dot{\phi} + I_{xy}(\dot{\theta}^2 - \dot{\phi}^2) + I_{xz}\dot{\theta}\,\dot{\psi} - I_{yz}\dot{\phi}\,\dot{\psi} = \tau_\psi \tag{3.10}$$

where m is the mass of the spinners, and $\{I_{xx}, I_{xy}, I_{xz}\ , I_{yy}, I_{yz}, I_{zz}\}$ are the moment of inertia of the robot in the local-body frame. τ_s is the spinners torque, and $\{\tau_\theta,\ \tau_\phi,\ \tau_\psi\}$ are the torques of the reaction wheels in the local-body frame. ν is the switch parameter between two modes of motion that is one in the CM and zero in the FM, and $S\left(z, \dot{z}\right)$ is the function of spring force that switches between two modes of motion expressed as:

$$S\left(z, \dot{z}\right) = \begin{cases} \frac{k_s\, z + b_s\, \dot{z}}{\cos\theta \cos\phi} - k_s\, L & \text{Contact Mode} \\ 0 & \text{Flying Mode} \end{cases} \tag{3.11}$$

In the Contact mode, the system will encounter impact force when the springs bottom out by reaching their maximum compression limit. This can be modeled as an elastic impact as follows:

$$\dot{z}^+ = -e_r\, \dot{z}^- \tag{3.12}$$

where the $\dot{z}^+$ and $\dot{z}^-$ are the after and before impact velocities respectively, and e_r is the impact coefficient of restitution [140, 141].

3.3.3 Controllability and Accessibility of the nonlinear system

In this section, we use the Lie-brackets theory to investigate the accessibility of the system. The mathematical framework for this analysis is based on the method presented in [111]. The full mathematical proof of this theory can be found in [29]. This method is briefly explained here. A control affine nonlinear system can be expressed as:

$$\dot{q} = f(q) + \sum_{i=1}^{m} g_i(q)\, u_i \tag{3.13}$$

where $q \in R^n$ are state variables, f is the drift vector field, g_i is the i^{th} input vector field and u_i is the control input. For this nonlinear system, starting from an initial state q_0 there exists a control that can drive the system to a space $H(q_0)$, which is formed by system vector fields and their nested Lie-brackets given as:

$$H = span\left(G^0, G^1, G^2, \cdots, G^m\right) \tag{3.14}$$

where

$$\begin{aligned} G^0 &\equiv \{g_1\} \\ G^1 &\equiv \{[g_1, g_2]\} \\ G^2 &\equiv \{[g_1, [g_2, g_3]]\} \\ &\vdots \\ G^m &\equiv \{[g_1, [g_2, [\cdots, g_m]]]\} \\ i &= 1, 2, \cdots, m; \end{aligned} \tag{3.15}$$

where the Lie-bracket of vector fields is defined as:

$$[X, Y] = L_X Y = Y\frac{\partial X}{\partial x} - X\frac{\partial Y}{\partial x} \tag{3.16}$$

The nonlinear system is fully Accessible if $\mathfrak{Rank}\ (span(L_o)) = n$, where $L_o \equiv [f, [G_0, [\cdots, G_m]]]$. We verify the accessibility of the system by using Eq. (3.15). For this purpose, Eqs. (3.4-3.10)

can be written in affine-control form as Eq. (3.13).

The matrix L_o assumes two forms, one for the *Flying Mode* (L_o^f) and the other for *Contact Mode* (L_o^c). The rank of L_o^f is found as 11 so the system is not controllable in the FM. The rank of L_o^c, however, is found 14, which means the system is controllable in the CM.

3.4 Control Design

The hopping robot is a hybrid dynamical system that has two dynamical modes; airborne and ground contact. From the controllability and accessibility section, we know that the system is fully controllable in the contact mode only. This will make the controller design complicated. In the CM, we can control all states while in the FM we can only control the orientation.

The control task for the robot is divided into two sub-tasks; jumping and tracking controls. The jumping control manipulates the spinners to achieve continuous jumps. Meanwhile, tracking control has also two tasks; stabilization and tracking. First, the stabilizing controller uses the flywheels to set the robot in an upright lateral position (θ control). The second controller also uses the flywheels such that the robot tracks a prescribed ground trajectory by regulating the angles ϕ and ψ.

Since we don't have direct control over position of the robot, the desired ground trajectory is mapped into the desired angles. Tracking the desired angles will cause the robot to land with specific orientations at given landing points. This way, the robot will track the desired ground trajectory. Mapping a desired trajectory into mapping angles generates a set of desired landing point on walking surfaces (points P_i in Fig. 3.2). Subsequently, the set of desired angles $\{\theta^d, \phi^d, \psi^d\}$ are found as given in Eqs. 3.17-3.19.

$$\theta^d = 0 \tag{3.17}$$

$$\phi_n^d = \cot^{-1}\left(\frac{2\,g\,h_n}{\sqrt{\dot{x}_n^2 + \dot{y}_n^2}}\right) \tag{3.18}$$

$$\psi_n^d = \tan^{-1}\left(\frac{\dot{y}_n}{\dot{x}_n}\right) \tag{3.19}$$

where h_n is the height of the each jump, and $\{x_n, y_n\}$ are the coordinates of the each contact point.

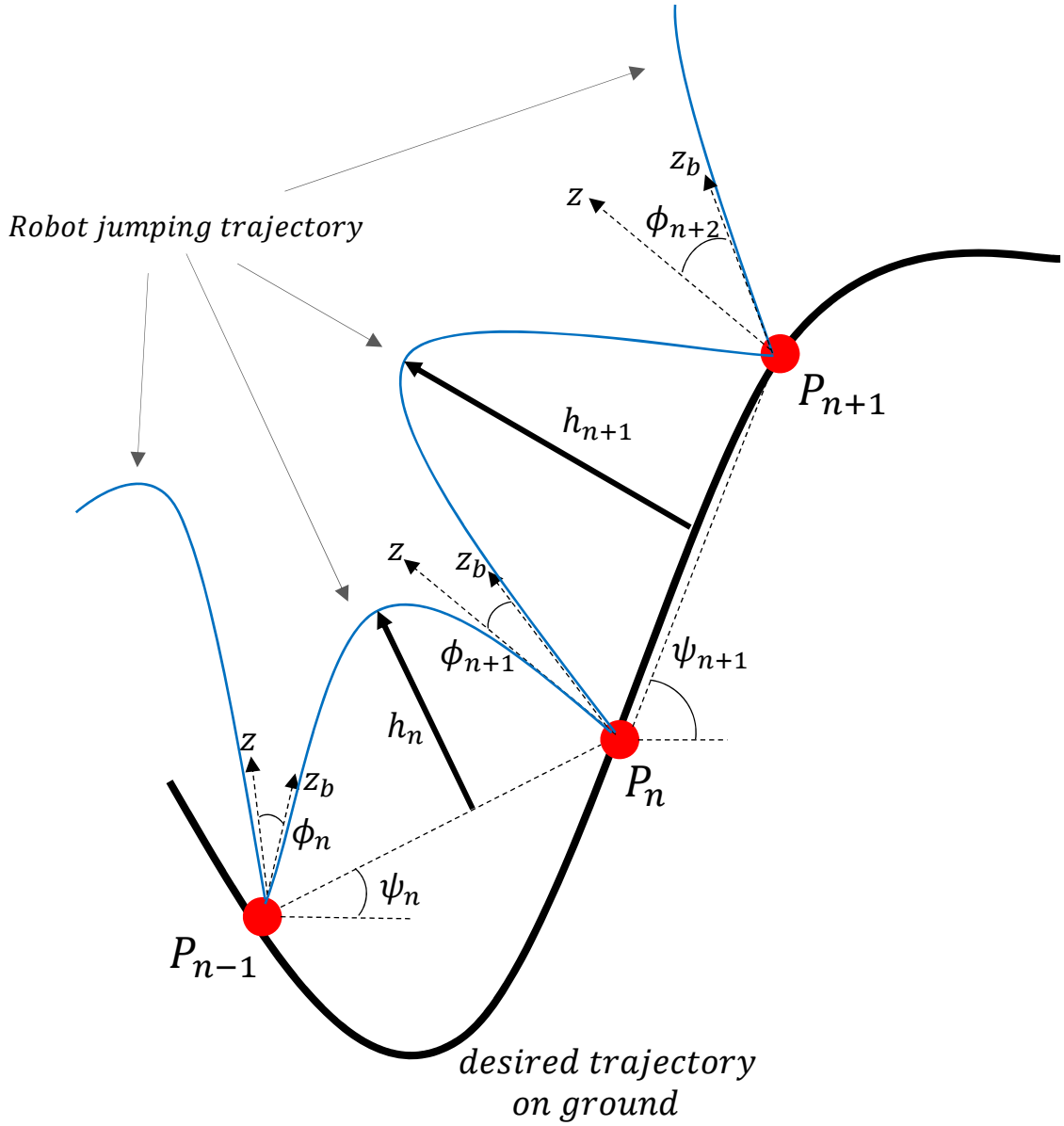


Figure 3.2: The desired path mapping. The desired trajectory is on the xy-plane.

Figure 3.3 depicts the control structure for the whole robot. The desired and measured angles $\{\theta, \phi, \psi\}$ are used in the tracking control block. In order to obtain the needed torques $\{\tau_\theta, \tau_\phi, \tau_\psi\}$ to achieve the required tasks, two different controllers are designed, SMC and SDRE. Simultaneously, jumping control block uses the measured $\{z, \alpha\}$ with the designed adaptive technique to produce the continuous jumps. Controller designs are explained in the upcoming subsections. The first part describes the enhanced adaptive position control

of the spinners discussed in [29] and the second part presents SMC and SDRE designs.

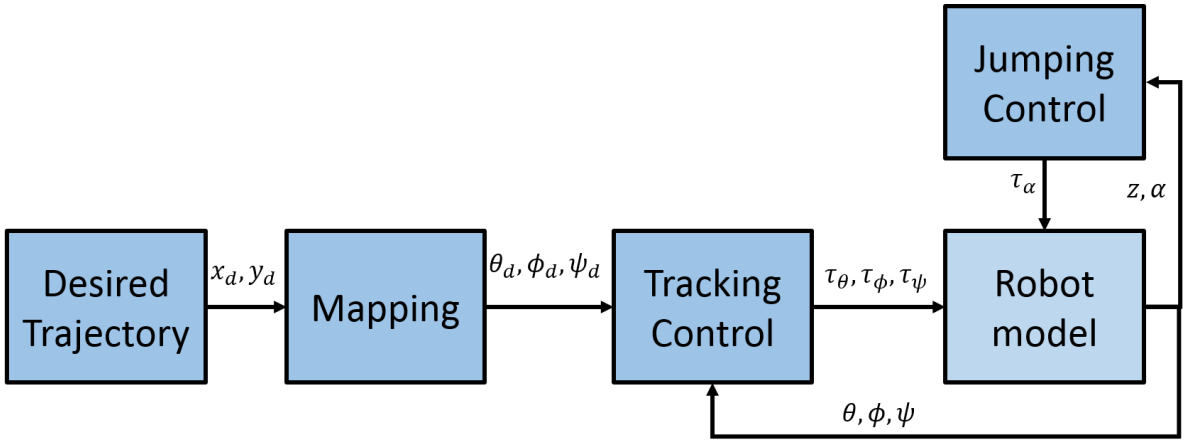


Figure 3.3: Control structure.

3.4.1 Jumping Control

By regulating position angles of the spinners, the robot can get the periodic actuation and this leads to continuous hopping. The main objective is reaching the specific angle of the spinners at the time the springs contact the ground. The spinners angles when the robot departures and contacts the ground are α_D and α_C, respectively. These two angles occur at the beginning and the end of the flight phase of the system. Starting from α_D, the controller adapts the angular velocity of the spinners during the flight phase to attain the desired α_C.

The heights of the center of mass in each jump at *departure time* (t_D) and *contact time* (t_C) are given by:

$$\begin{aligned} z_C &= \frac{-2mL\cos\alpha_C}{2m+M_T} + H_L \\ z_D &= \frac{-2mL\cos\alpha_D}{2m+M_T} \end{aligned} \tag{3.20}$$

where H_L is the different height departure and contact points of the robot in each jump.

The velocity of the robot in the *departure time* can be written as follows:

$$\dot{z}_D = \frac{M_T \dot{z}_i + 2m\left(\dot{z}_i + L\,\omega \sin\alpha_D\right)}{2m + M_T} \tag{3.21}$$

where $\dot{z}_i$ is the velocity of the robot at that time.

The duration of the FM (Δt_f) can be obtained from solving the following quadratic equation:

$$z_C - z_D = \frac{1}{2} g (\Delta t_f)^2 + \dot{z}_D \Delta t_f \tag{3.22}$$

The adaptation law to obtain the desired angular position of the spinners is given by:

$$\begin{aligned} \alpha_d(t) &= \omega(t - t_D^{(i)}) + \alpha_D^{(i)} \\ \omega &= \frac{\alpha_p^{(i)}}{\Delta t_f} \end{aligned} \tag{3.23}$$

where $t_D^{(i)}$ is represented as the i^{th} total *departure time* and the $\alpha_p^{(i)}$ is the i^{th} progression angle. By this adaptive control, the angle of the spinners in *contact time* remains constant.

3.4.2 Tracking Control

The design processes of each controller are briefly presented in the following sections.

3.4.2.1 Sliding Mode Control

In the previous section, the adaptive controller is designed to control two state variables $\{z, \alpha\}$. In the following, SMC is applied to control the state variables $\xi = \{\theta, \phi, \psi\}$. By controlling them, the robot will stabilize at the desired angles as dictated the desire trajectory. From equations of motion Eqs. 3.4-3.10, the state variable vector is redefined as:

$$\ddot{\xi} = F(\xi\,, \dot{\xi}) + \bar{G}\,u \tag{3.24}$$

where u is the control input vector containing of the torques applied by the reaction wheels. In the sliding mode control approach, a surface $s(t)$ is defined as a function of tracking errors. The controller ensure that the system trajectory asymptotically tend to the surface.

Subsequently, the trajectory would not leave the $s(t)$ (sliding motion) for the remainder of the control action. In addition, when st equal to zero, the states converge to desired values. The tracking error vector of the state variables $\tilde{\xi}(t)$ and a linear time-varying sliding surface in terms of tracking error vector are defined as follows:

$$\begin{aligned} \tilde{\xi}(t) &= \xi(t) - \xi_d(t) \\ s(t) &= \dot{\tilde{\xi}}(t) + \lambda_F \tilde{\xi}(t) \end{aligned} \tag{3.25}$$

where λ_F is a diagonal positive definite matrix. Consider a Lyapunov function as:

$$V = \frac{1}{2} s(t)^T s(t) \tag{3.26}$$

The time derivative of the Lyapunov function by substituting Eqs. (3.24), (3.25) and (3.26) is given by:

$$\dot{V} = s(t)^T (F(\xi\,, \dot{\xi}) + \bar{G}\, u - \ddot{\xi}_d(t) + \lambda_F \dot{\tilde{\xi}}(t)) \tag{3.27}$$

The stability and reaching condition of $s(t)$ is defined as [142]:

$$\dot{V} < -\eta^T |s(t)| \tag{3.28}$$

where η is a constant vector with positive elements and determines how fast the trajectory converges to the surface. The control law is:

$$U_{con} = \bar{G}^{-1} \left[-F(\xi\,, \dot{\xi}) + \ddot{\xi}_d(t) - \lambda_F \dot{\tilde{\xi}}(t) - K_{con}\, \mathrm{sgn}(s(t)) \right] \tag{3.29}$$

Using the sign function in Eq. (3.29) induces chattering into control action. We use a saturation function instead. In this way, the sign function is replaced with a continuous saturation function, which yields:

$$\mathrm{sat}\left[\frac{s(t)}{\Phi}\right] = \begin{cases} \frac{s(t)}{\Phi} & |\frac{s(t)}{\Phi}| < 1 \\ \mathrm{sgn}(s(t)) & |\frac{s(t)}{\Phi}| \geq 1 \end{cases} \tag{3.30}$$

where Φ is the boundary layer thickness of the saturation function. By applying these control

inputs to reaction wheels, the attitude and forward motions of the robot are controlled.

3.4.2.2 SDRE control

Next, SDRE solution for the system is examined. The solution of a nonlinear control problem using SDRE method consists of the following steps:

1. Define a cost function.
2. Express the dynamics in state-dependent coefficient form.
3. Solve the resulting LQR equation.
4. The controller is obtained by using the traditional LQR type formula, with all the coefficients depending on the states instead of the time.

The cost function used in the solution of the benchmark problem is:

$$J = \frac{1}{2}\int_0^\infty [x^T\, Q(x)\, x + u^T\, R(x)\, u] \tag{3.31}$$

where $Q(x)$ and $R(x)$ are positive definite matrices. Equation (3.24) is written in state-dependent coefficient form as follows:

$$\dot{x} = A(x)\, x + B(x)\, u \tag{3.32}$$

For the purpose of obtaining SDRE solution the state dependencies in the matrices $A(x)$, $B(x)$, $Q(x)$, and $R(x)$ are ignored, and the resulting optimal control problem is solved for each value of the states. The solution is obtained by solving the Riccati equation for each value of state as follows:

$$\begin{aligned} &A^T(x)\, P(x) + P(x)\, A(x) \\ &- P(x)\, B(x)\, R^{-1}(x)\, B^T(x)\, P(x) + Q(x) = 0 \end{aligned} \tag{3.33}$$

The resulting Riccati equation solution is a function of the states. The feedback controller

is:

$$u = -R^{-1}(x)\, B^T(x)\, P(x)\, x \tag{3.34}$$

The resulting controller has the same structure as the LQR controller, except that all the coefficients are state dependent. The SDRE controller, by its construction, ensures that there is a near optimal solution for the system.

The choice of the matrix $A(x)$ is not unique. One of the conditions is that in the domain of interest the pairs $(A(x), B(x))$ and $(A(x), Q^{1/2}(x))$ should be pointwise controllable and detectable, respectively.

These conditions could be easily verified by checking the rank of the controllability and Observability matrices. A sufficient way to test stabilizability condition is checking the controllability matrix :

$$\mathbf{C} = \left[B \,|AB\, |A^2B\, |\cdots|\, A^{n-1}B\right] \tag{3.35}$$

Which has $\mathfrak{Rank}\,[\mathbf{C}] = n$. Similarly, a sufficient way to test detectability is that the observability matrix

$$\mathbf{O} = \left[Q^{1/2}\, |Q^{1/2}A\, |Q^{1/2}A^2\, |\cdots|\, Q^{1/2}A^{n-1}\right] \tag{3.36}$$

has $\mathfrak{Rank}\,[\mathbf{O}] = n$. This can be guaranteed by ensuring that $Q(x)$ is positive-definite.

There exists an infinite number of SDC parameterization that yields different controllability matrix. Cloutier and Stansbery [70] suggested a method to obtain the matrix $A(x)$. They showed that if $\dot{x}_i$ depends on the x_j state then the corresponding element in the $A(x)$ matrix should be non-zero.

Also, for these type of multi-variable problems, x has at least two components, x_1 and x_2. Note that there is a single scalar nonlinear term $f_1(x)$ appearing in one of the state equations. In that state equation, one parameterization has the nonlinearity $\frac{f_1(x)}{x_1}$ appearing as a coefficient of x_1. While a second parameterization has the nonlinearity $\frac{f_1(x)}{x_2}$ appearing as a coefficient of x_2 . Thus, there always exists at least two parameterizations. Suppose $A_1(x)$ and $A_2(x)$ are two distinct SDC parameterizations, such that $f(x) = A_1(x)x = A_2(x)x$. Then $A(x, \beta)$ from Eq. (3.37) is also an SDC parameterization for any β.

$$A(x, \beta) = \beta A_1(x) + (1 - \beta)A_2(x) \tag{3.37}$$

SDRE method produces a close loop solution that is asymptotically stable. The close loop solution becomes $\dot{x} = A_{CL}(x)\, x$, where $A_{CL}(x)$ is the close loop SDC matrix given by:

$$A_{CL}(x) = A(x) - B(x)\, K(x) \tag{3.38}$$

where $k(x) = R^{-1}(x)\, B^T(x)\, P(x)$. Applying the Mean Value Theorem to $A_{CL}(x)$ gives:

$$A_{CL}(x) = A_{CL}(0) + \frac{\partial A_{CL}(\hat{x})}{\partial x} x \tag{3.39}$$

where $\frac{\partial A_{CL}(\hat{x})}{\partial x}$ generates a tensor, and the vector $\hat{x}$ is that point on the line segment joining the origin 0 and x . By substituting Eq. (3.39) in Eq. (3.32), we have:

$$\dot{x} = A_{CL}(0)\, x + x^T \frac{\partial A_{CL}(\hat{x})}{\partial x} x \tag{3.40}$$

which gives:

$$\dot{x} = A_{CL}(0)\, x + \Psi(x, \hat{x})\, \|x\| \tag{3.41}$$

where $\Psi(x, \hat{x}) \equiv \frac{1}{\|x\|} x^T \frac{\partial A_{CL}(\hat{x})}{\partial x} x$, such that

$$\lim_{\|x\| \to 0} \Psi(x, \hat{x}) = 0.$$

Hence, in a neighborhood around the origin, the linear term which has a constant stable coefficient matrix $A_{CL}(0)$ dominates the higher-order terms, yielding local asymptotic stability.

We develop a mathematical approach to form a SDRE controller for the system. The system defined in section 3.3 is given by Eqs. (3.8)-(3.10). Since the condition $f(0) = 0$ is satisfied, the system can be transformed into the SDC form given in Eq. (3.32). Following the rule specified above to pick up the plant matrices for the orientation angles, we can get

the parameterization. Equations (3.8)-(3.10) are redefined as:

$$\begin{aligned}
\ddot{\phi} &= f_1(\dot{\phi}, \dot{\theta}, \dot{\psi}) + g_1(\tau_\theta, \tau_\phi, \tau_\psi) \\
\ddot{\theta} &= f_2(\dot{\phi}, \dot{\theta}, \dot{\psi}) + g_2(\tau_\theta, \tau_\phi, \tau_\psi) \\
\ddot{\psi} &= f_3(\dot{\phi}, \dot{\theta}, \dot{\psi}) + g_3(\tau_\theta, \tau_\phi, \tau_\psi)
\end{aligned} \tag{3.42}$$

One can find the details of Eq. (3.42) and SDC form of the equations of orientation angles in the appendix. Now, since we have obtained SDRE parameterization of the system, we implement the control law on the jumper robot to regulate the orientation angles.

3.5 Stability Analysis

In this section, we analyze the stability of the system in neighborhoods of equilibria. In this regard, we use *Poincare Map* and *Floquet Theory*. Since closed form analytic solution of differential equations of motion is not available, numerical integration is used [47].

Let $q(t)$ be the the periodic solution of the equations of motion for all states. The simplified hyper plane of the Poincare section can be defined as:

$$\begin{aligned}
&\mathfrak{S} = (z, \theta, \dot{\theta}, \phi, \dot{\phi}, \alpha, \dot{\alpha}) \in R^7 \\
&x = y = t, \dot{x} = \dot{y} = 1, \dot{z} = \dot{\psi} = 0, \psi = 45^\circ
\end{aligned} \tag{3.43}$$

If $q[k] \in \mathfrak{S}$ denotes the k^{th} intersection of $\mathfrak{S}$ by the flow of $q(t)$ the discrete-time Poincare Map $\mathfrak{P} : \mathfrak{S} \to \mathfrak{S}$ can be expressed as:

$$q[k+1] = \mathfrak{P}(q[k]) \tag{3.44}$$

Subsequently, if q^* stands for fixed point of the Poincare Map, then the local exponential stability of q^* on $\mathfrak{S}$ is equivalent to local exponential stability of the underlying limit cycle. We used Floquet Theory for stability analysis of limit cycles [143, 144]. In this regard, the

local linearizion of the Poincare Map about its fixed point yields

$$\begin{aligned} \mathfrak{P}(q) &\simeq J(q^*)(q - q^*) \\ J &= \frac{\partial \mathfrak{P}(q)}{\partial q} \end{aligned} \tag{3.45}$$

where J is 7×7 linearized Jacobian matrix of $\mathfrak{P}$

$$J = [J_{i,j}]; J_{i,j} = \frac{\partial \mathfrak{P}}{\partial q_i} \tag{3.46}$$

Then, the floquet or characteristic multipliers ρ_i's are defined as eigenvalues of the Jacobian matrix J

$$\rho_i = \|\lambda_i\| \, ; \lambda_i = \text{eig}(J) \tag{3.47}$$

where ρ_i and λ_i are the i^{th} characteristic multiplier and Jacobian matrix eigenvalue, respectively, and "$\| \, \|$" is the modulus operator. Therefore, the limit cycle stability can be identified by Floquet multiplier metrics as:

$$q(t) = \begin{cases} \text{stable} & \forall \, \|\rho_i\| < 1 \\ \text{marginally stable} & \forall \, \|\rho_i\| = 1 \\ \text{unstable} & \forall \, \|\rho_i\| > 1 \end{cases} \tag{3.48}$$

Since the equations of motion of the hopper is different from [129], we conduct numerical simulation examples to show the limit cycle behavior of the robot and find the stability of the system. The experimentally estimated prototype parameters in Table 3.1 also used for the numerical simulation.

As Kashki *et al.* [129] showed, the jumping motion converge to a limit cycle. Using the adaptive scheme, $\dot{\alpha}$ is controlled to produce continuous jumps, as can be seen in Figs. 3.4-3.5. The continuous jumps can be described as a limit cycle of the system, different initial conditions are tested, and all of them converge to the same limit cycle as shown on the phase plane of $z - \dot{z}$ in Fig. 3.6. Here, we can use the *Poincare Map* and *Floquet Theory*. The Floquet multipliers associated with the corresponding Poincare map for the simulations are

Table 3.1: The detail of the parameters in simulation.

parameter	value	parameter	value
M_T	4 Kg	I_{xx}	0.11 Kg.m^2
m	0.2 Kg	I_{yy}	0.14 Kg.m^2
L_{sp}	0.065 m	I_{zz}	0.19 Kg.m^2
L	0.125 m	I_{xy}	0.08 Kg.m^2
k_s	2000 N/m	I_{xz}	0.02 Kg.m^2
b_s	5 N.s/ m	I_{yz}	0.02 Kg.m^2
λ	0.1	K_{con}	10

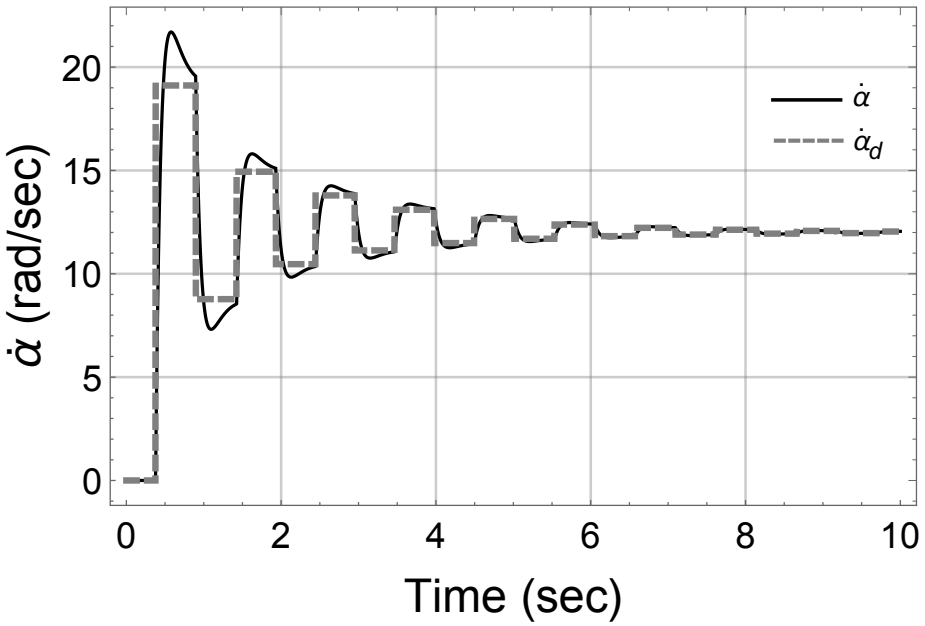


Figure 3.4: Time history of $\dot{\alpha}$ for the single ground level.

calculated numerically as:

$$\rho_1 = \|\lambda_1\| = 0.3866 < 1 \; ; \;\; \rho_2 = \|\lambda_2\| = 0.00374 < 1$$
$$\rho_3 = \|\lambda_3\| = 0.0025 < 1 \; ; \;\; \rho_4 = \|\lambda_4\| = 0.00012 < 1$$
$$\rho_5 = \|\lambda_5\| = 0.0002 < 1 \; ; \;\; \rho_6 = \|\lambda_6\| = 3 \times 10^{-6} < 1$$
$$\rho_7 = \|\lambda_7\| = 0 < 1 \tag{3.49}$$

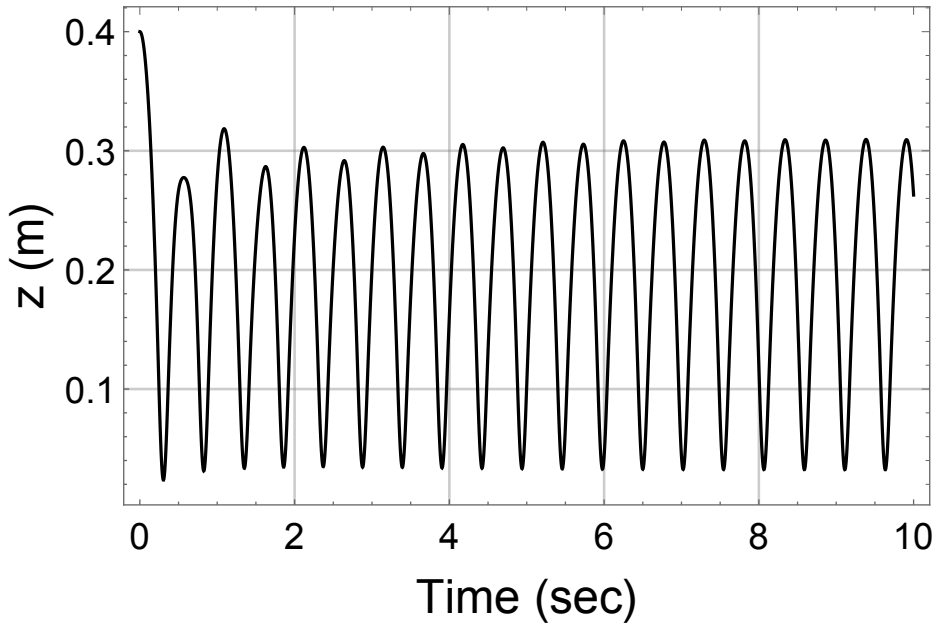


Figure 3.5: The height of the robot for single ground level.

All norms of characteristic multiplier of the system are less than one, which means that the limit cycle is asymptotically stable. Also, numerical simulations show that different limit cycles can be obtained for different contact angles adaptive regulators (α_c). A set of limit cycles corresponding to different contact angle are depicted in Fig. 3.7. As one can see, the maximum jumping height of the robot is estimated to be at 31.4°.

Also, we make the analysis on the effect of the spinner and sagittal angles on the jumping height and horizontal range. The jumping motion robot can be described as a projectile motion. So, the maximum jumping height and horizontal range are taken into account as important metrics. Increasing the sagittal angle decreases the jumping height while increasing the horizontal range.

Finding the optimum angles to achieve the maximum jumping height and range is critical. But increasing the sagittal angle requires more friction to prevent slippage. The slip condition is given by:

$$\frac{F_x}{F_y} > \mu \tag{3.50}$$

where F_x and F_y are the horizontal and vertical reaction forces exerted at the ground contact

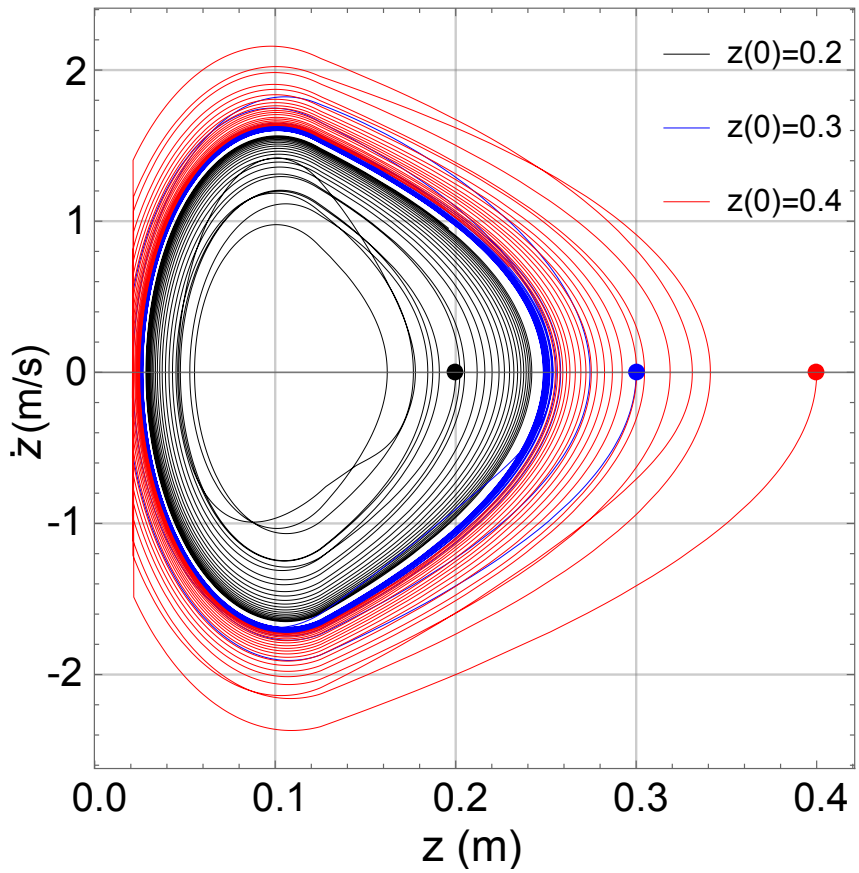


Figure 3.6: Limit Cycles of the jumping motion.

point respectively, and μ is the friction coefficient. In our system this condition is rewritten as:

$$\frac{S\left(z,\dot{z}\right)\sin\phi\cos\theta}{M_T\,g+S\left(z,\dot{z}\right)\cos\phi\cos\theta}>\mu \tag{3.51}$$

Figure 3.8 depicts the variation of ϕ with respect coefficient of friction μ. As one can see, the following general trend is observed by carrying out the numerical analysis:

$$\mu_1<\mu_2\Leftrightarrow\phi_{n1}<\phi_{n2} \tag{3.52}$$

Without loss of generality, we assume that $\mu=0.2$, which results in $\phi_n=10^\circ$. Now, we can find the effect of the spinner and sagittal angles on the jumping height and horizontal range. Figures 3.9-3.10 depict the maximum height and range in terms of different values of α_c and ϕ. The maximum height and range occur at $\alpha_c=43^\circ$ and $\phi=10^\circ$. Figure 3.9

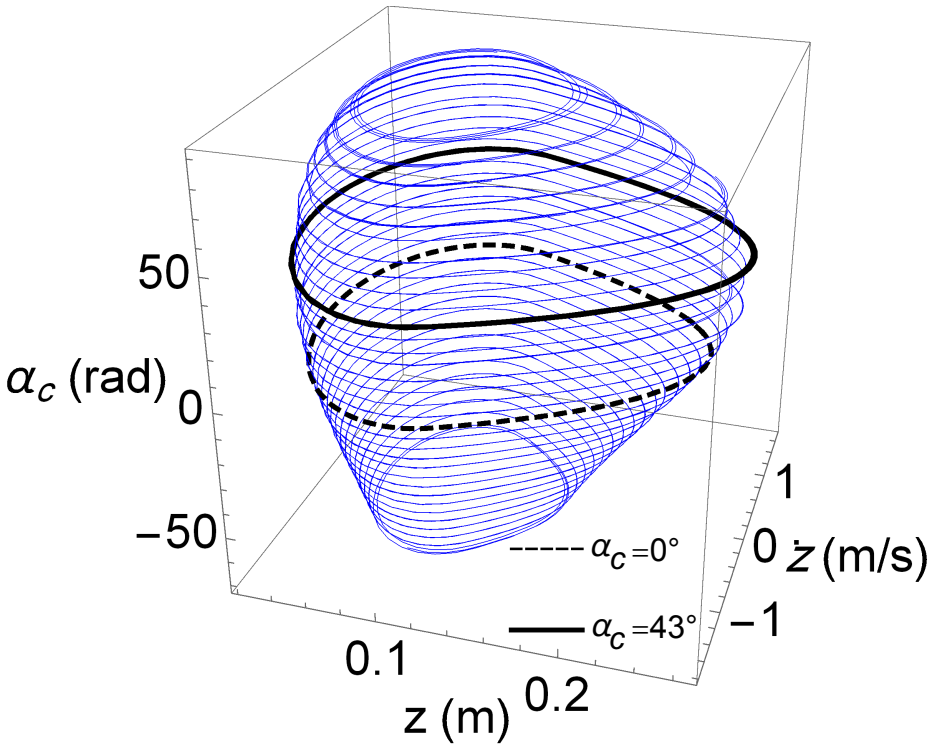


Figure 3.7: Limit cycles corresponding to different values of contact angles.

shows that the maximum jumping height depends on α only.

Now, we investigate the effect of α_C on jumping motion of the robot on stairs. The jumping motion is regulated by the spinners as we explained in 3.4.1. Our primary focus is on the jumping height, range, and stability. Figure 3.11 depicts the variation of the maximum stably achievable step height with respect to spinner's angle, while other parameters are fixed. Here, we let the stair depth be 15 cm. The figure shows a drop for $\alpha_C > 40$. This drop is due to the contraction observed in the sizes of the limit cycles in upper hemisphere of Fig. 3.7.

Figure 3.12 also shows that the effect of α_C on maximum stair depth, which the robot has stable jumps. Here, we let the stair height be 5 cm. The maximum stair depth will increase with increasing α_C.

3.6 Simulation Results

This section presents numerical simulation of two scenarios to illustrate the performance of the proposed control methods. Simulations were conducted using Wolfram Mathemat-

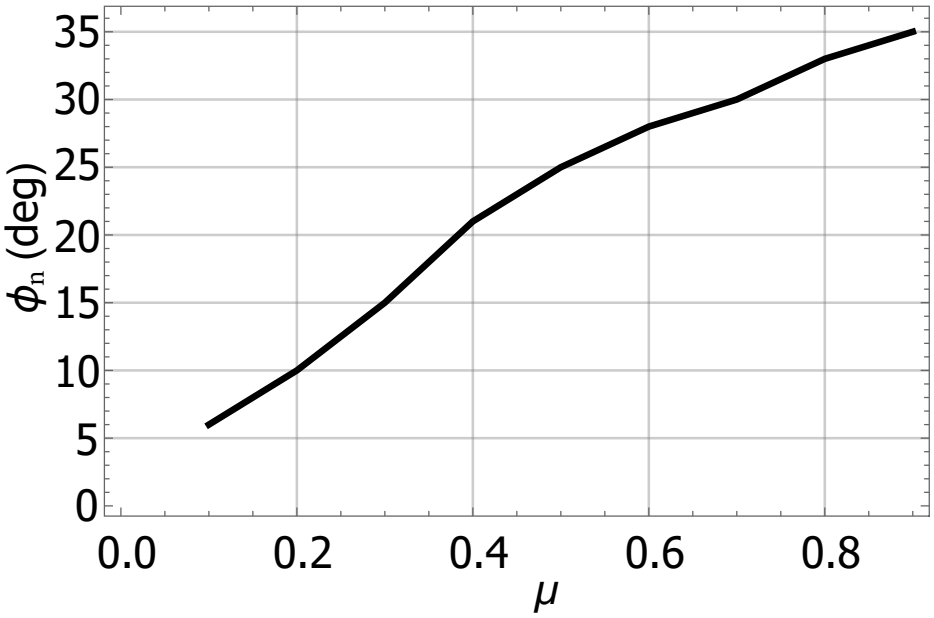


Figure 3.8: No-slipping sagittal angle value for different value of μ.

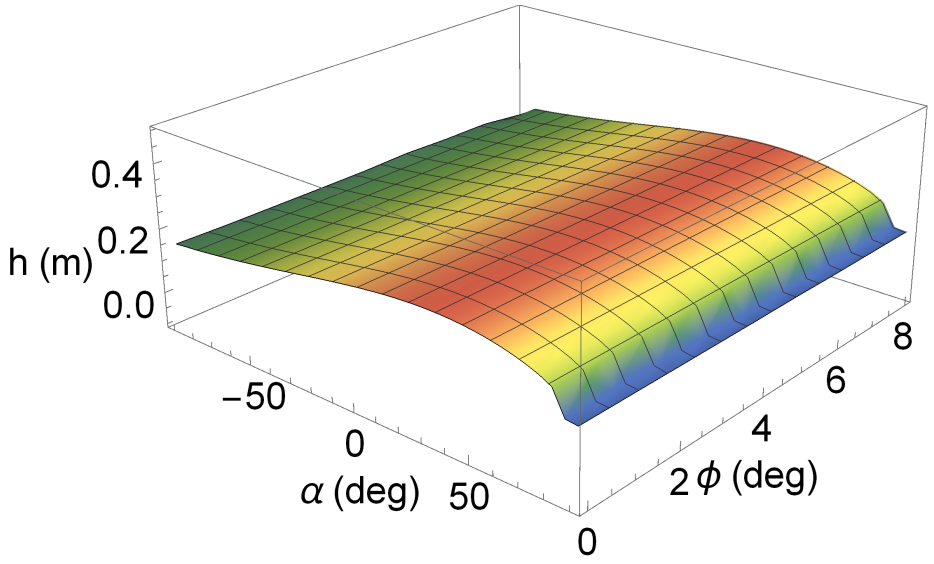


Figure 3.9: Jumping height for different values of the α_c and ϕ.

ica software. As presented before, adaptive control is used to control the states α and z. Meanwhile, SMC and SDRE controllers are used to control the states θ, ϕ, and ψ. Finally, the states x and y are controlled indirectly through the other states α, z, θ, ϕ, and ψ. The

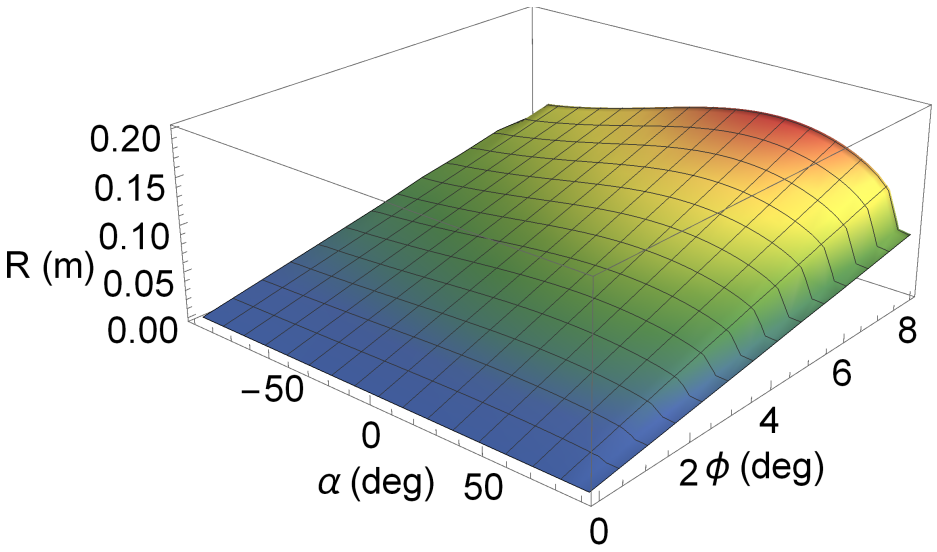


Figure 3.10: Jumping horizontal range for different values of the α_c and ϕ.

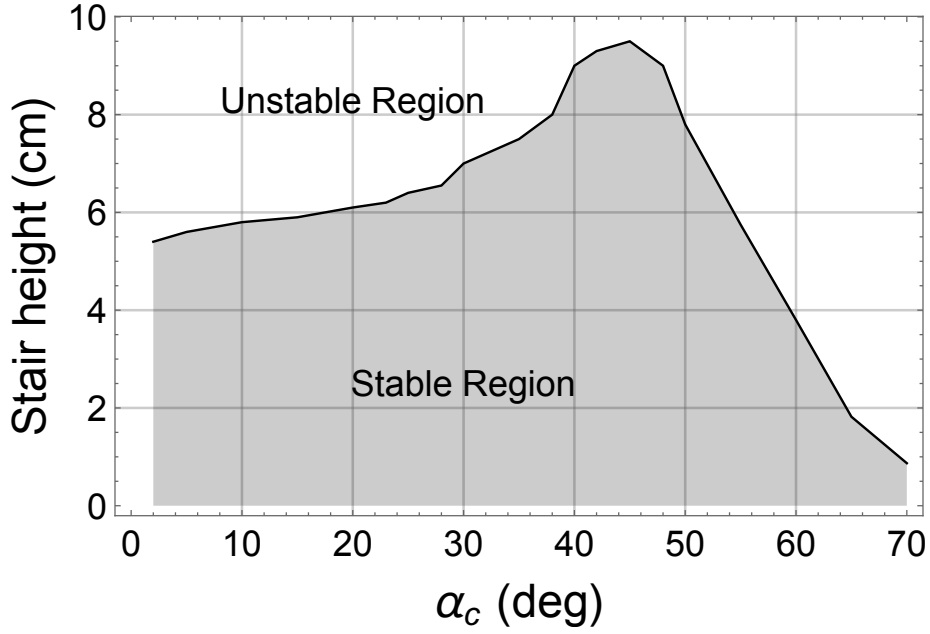


Figure 3.11: Variation of maximum stair height for different values of α_c.

initial conditions are selected as $x(0) = y(0) = 0, z(0) = 0.4; \phi(0) = 0.1\,\text{rad}$, $\theta(0) = 0.05\,\text{rad}$, and $\psi(0) = 0.84\,\text{rad}$. We use two different measurement indicators to evaluate the quality of proposed controller's responses. These indicators are: Integral Absolute Error (IAE) and

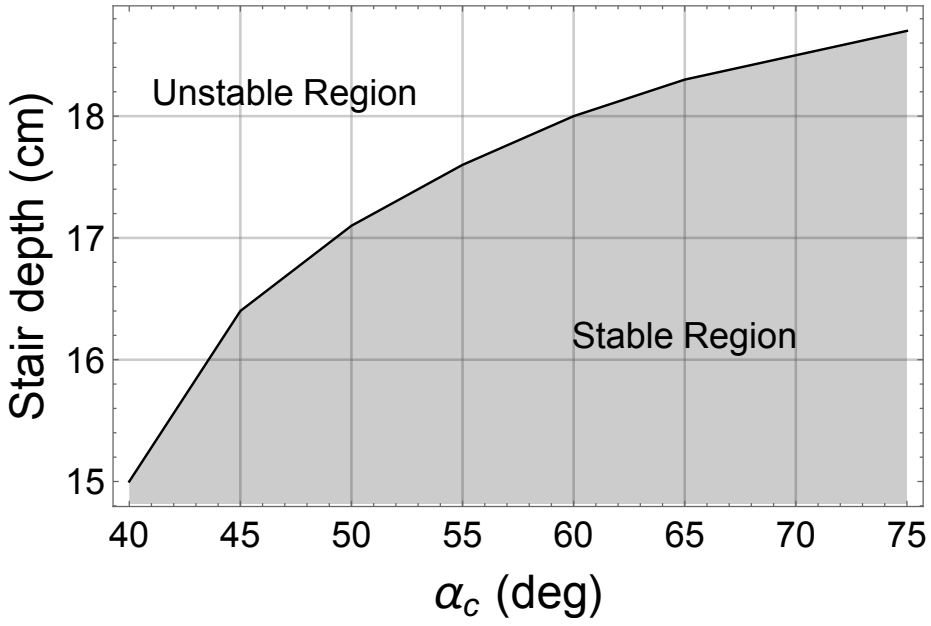


Figure 3.12: Variation of maximum stair depth for different values of α_c.

Integral Time Absolute Error (ITAE). These measurements can be calculated as follows:

$$IAE_q = \int_0^{t_{sim}} \left| q - q^d \right| dt \tag{3.53}$$

$$ITAE_q = \int_0^{t_{sim}} t \left| q - q^d \right| dt \tag{3.54}$$

$$q = \{\theta, \phi, \psi\}$$

where t_{sim} is the total simulation time, q is the states, and q^d is the desired value for the states.

3.6.1 Scenario 1: Stairs

Now we present the example for stair motion. We assume that the stair's height and depth are 5 and 18 cm respectively, which results in $\phi = 5°$. For this scenario, a comparison between two controllers are shown in Fig. 3.13. As can be seen, both controllers converge to the desired values. In one hand, SDRE has optimal properties, so it converges with minimum cost. On the other hand, SMC has robust properties, so it converges even when parameter uncertainties are present. Table 3.2 indicates the performance measurements

of two proposed controllers on the system. In this case, we assume that there are not any uncertainties on the system. As one can see, SDRE performs better than SMC. It has faster convergence time as well as lower tracking errors. It regulates the system with minimum state error and control efforts (Eq. 3.31).

Figures 3.14-3.15 show the regulated $\dot{\alpha}$ and jumping height on stairs. Figure 3.14 depicts that jumping on stairs required different angular velocity of spinners to regulate continuous jumps. One can see this jumps on stairs in Fig. 3.15.

Table 3.2: Performance measurements of two proposed control schemes for scenario 1.

Controllers	States	IAE	ITAE
SMC	θ	0.0429	0.00846
	ϕ	0.0918	0.01722
	ψ	3×10^{-6}	35×10^{-6}
SDRE	θ	0.0074	5×10^{-5}
	ϕ	0.0167	12×10^{-5}
	ψ	1×10^{-6}	14×10^{-6}

3.6.2 Scenario 2: Sinusoidal Path

To demonstrate the maneuverability of the robot, we force it to track a sinusoidal path on the ground. The initial value of the states are identical to the ones chosen in scenario 1. In this scenario, we simulate two different conditions. At first, we conduct the simulation without any uncertainties. The results of this assumption are shown in Fig. 3.16. In second situation, the system is assumed to include the uncertainties on the moment of the inertia of the robot. Figure 3.17 depicts the simulation results for this case.

3.6.2.1 Without uncertainty

As one can see, two states ϕ and θ have the same tracking conditions shown in Figs. 3.13a-3.13b. Figure 3.16c shows the time history of the state ψ, which tracks the sinusoidal path. Figure 3.16d demonstrates the maneuverability of the robot in tracking a complex path in

the 3D environment. The maximum height $h = 31cm$ is the same for all scenarios, and it is independent of ψ. As we expect, similar to scenario 1, SDRE controller outperforms SMC in the absence of uncertainties.

3.6.2.2 With uncertainty

For this case, SDRE controller becomes unstable as can be seen in Fig. 3.17. As we explained in 3.4.2.2, SDRE control law depends entirely on system's parameters. Thus, uncertainties in system's parameters lead to unstable control action. Meanwhile, SMC control can easily cope with uncertainties as can be seen in Fig. 3.17. Table 3.3 presents the values of the performance measurements of the two control methods.

Table 3.3: Performance measurements of two proposed control schemes for scenario 2.

Conditions	Controllers	States	IAE	ITAE
Without uncertainties	SMC	θ	0.0429	0.00846
		ϕ	0.0918	0.01722
		ψ	3×10^{-6}	35×10^{-6}
	SDRE	θ	0.0074	5×10^{-5}
		ϕ	0.0167	12×10^{-5}
		ψ	1×10^{-6}	14×10^{-6}
With uncertainties	SMC	θ	0.1036	0.2871
		ϕ	0.1842	0.3048
		ψ	0.2005	0.3487
	SDRE	θ	∞	∞
		ϕ	∞	∞
		ψ	1.459	4.358

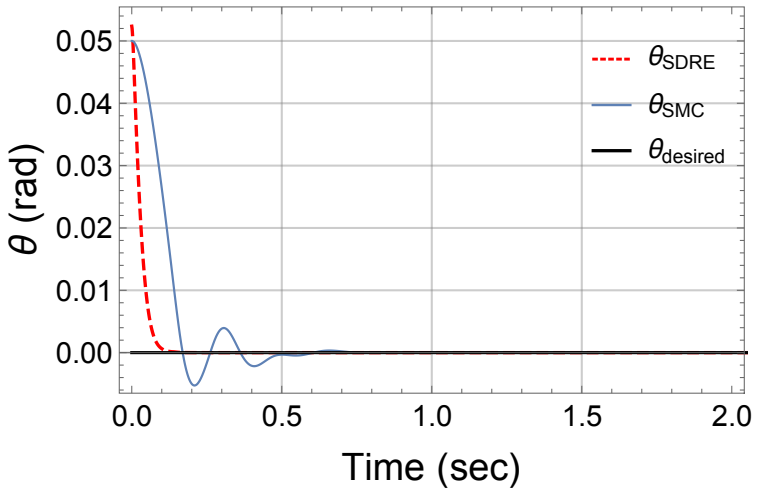
3.7 Discussion of the Results

Here, we discuss the limitation of proposed system. We assumed that the springs are linear. In general, however, the springs may exhibit nonlinear behaviors. Also, we assumed that the contact points of the springs with the ground can be treated points. But, in reality the contacts take place over elliptical region. In addition, the proposed control methods

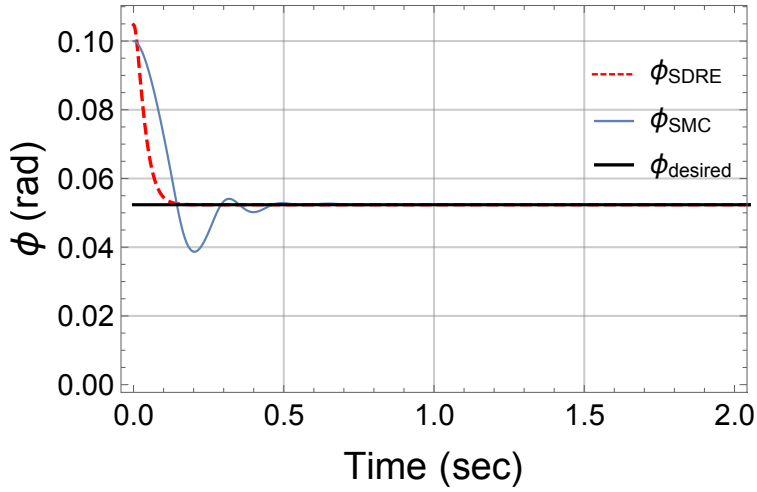
have their own limitations. Although SMC has robust properties, its response is not optimal. SDRE has optimal properties, but it is not able to overcome the presence of the uncertainties and disturbances. As one can see, because of the optimal properties of the SDRE, the convergence speed of the response of SDRE is faster than that of the SMC. However, in the presence of uncertainties, SMC regulates the system and SDRE exhibits unstable behavior. Moreover, the proposed controllers are theoretically capable of rejecting external disturbance.

3.8 Conclusion

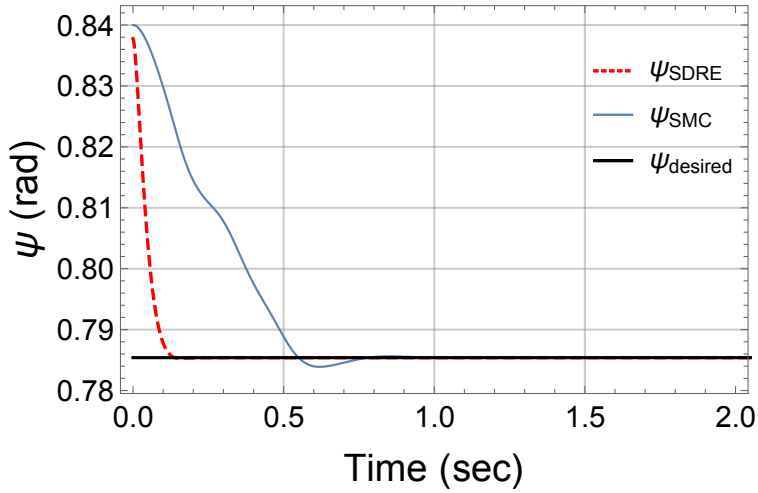
This section presented a unique inertial actuation hopping robot that can navigate in the 3D environment. Rotating mass spinners acted as inertial actuators to generate the jumping force while flywheels were used to stabilize and control the orientation of the robot. The robot has two modes of motion, flying and contact modes. In the contact mode, the adaptive controller regulated the spinners angle to generate continuous jumps. Also, Tracking controllers were used to stabilize the robot in both modes and change the orientation of the robot in flying mode. We proposed two tracking controllers, SMC and SDRE approaches. The stability analysis was conducted using Poincare Map and Floquet Theory. The parametric analysis over the effects of the spinner's and sagittal angles on the maximum jumping height and range was conducted. Finally, two simulation scenarios validated the performance the two tracking controllers and were shown the comparison between them. In our system, SDRE performed better than SMC. It had faster convergence time and lower tracking errors when no uncertainties are presents. The stairs scenario showed the performance of the tracking controllers on the adjustment between jumping height and range. We modified the adaptive controller to regulate the jumping force on different ground levels. The sinusoidal scenario demonstrated the maneuverability of the robot. SDRE performed optimally when no uncertainties are presents. Whereas, SMC outperforms in the presents of the uncertainties.



(a)



(b)



(c)

Figure 3.13: Orientation angles for jumping on stairs.

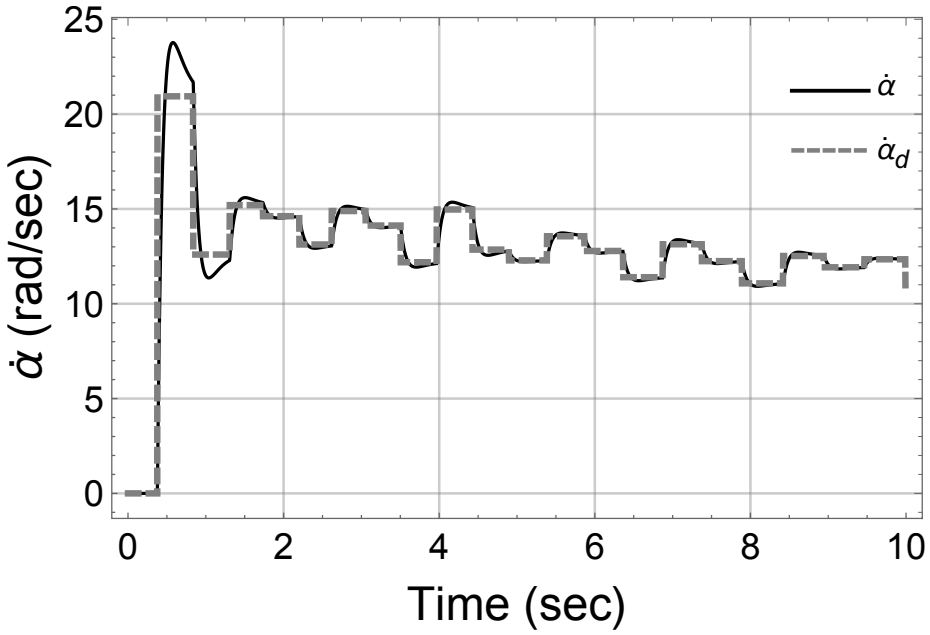


Figure 3.14: Time history of $\dot{\alpha}$ for the stairs scenario.

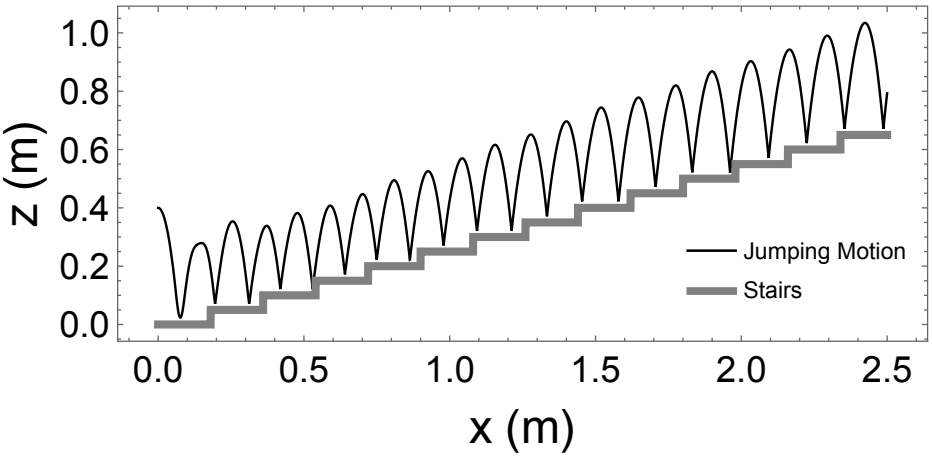


Figure 3.15: Jumping motion on stairs.

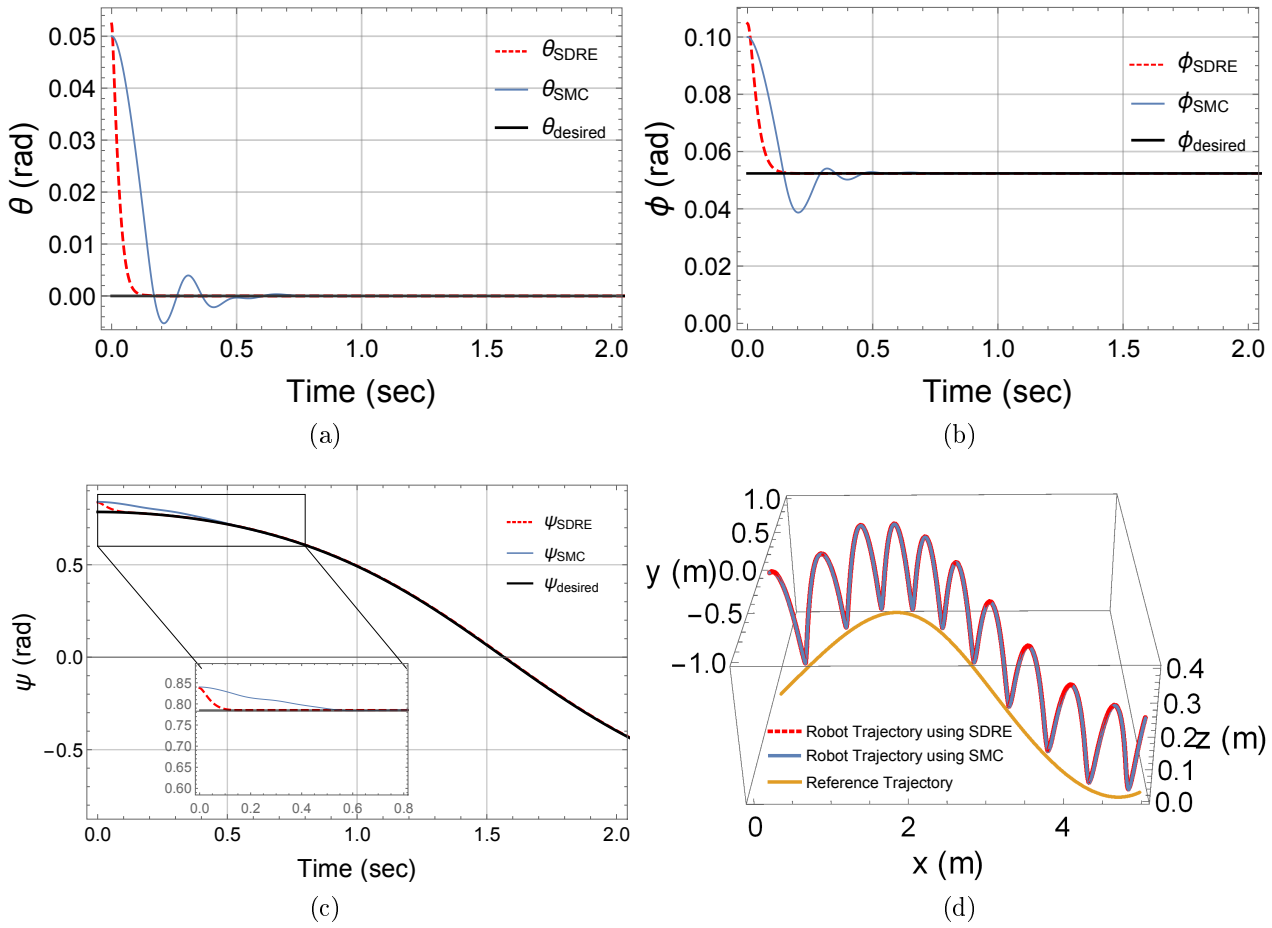
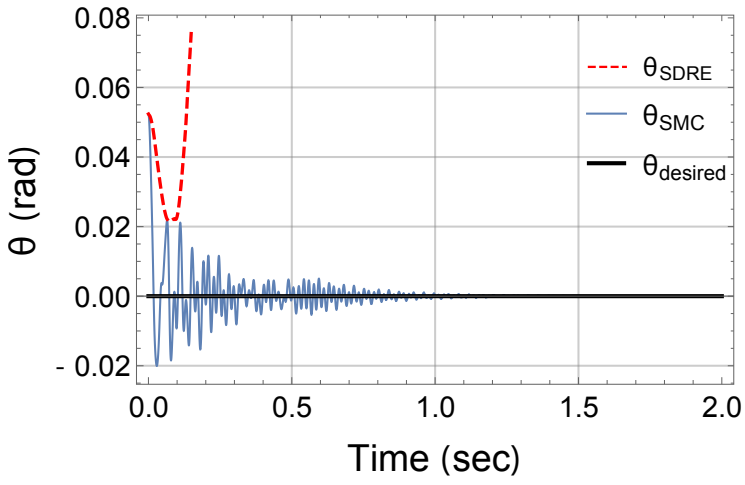
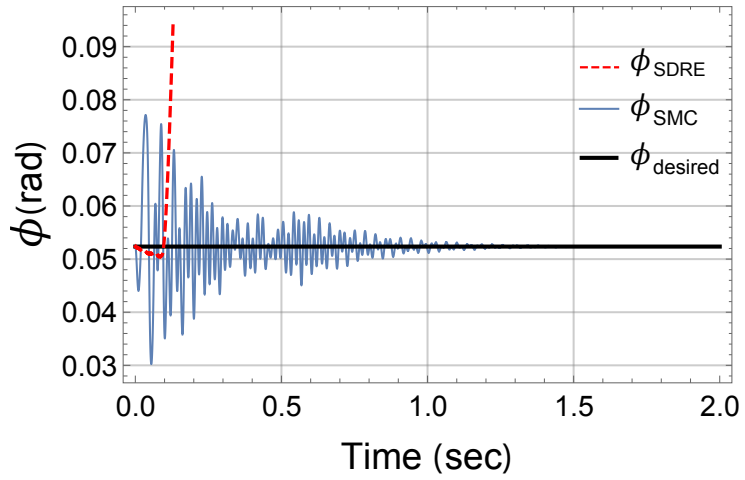


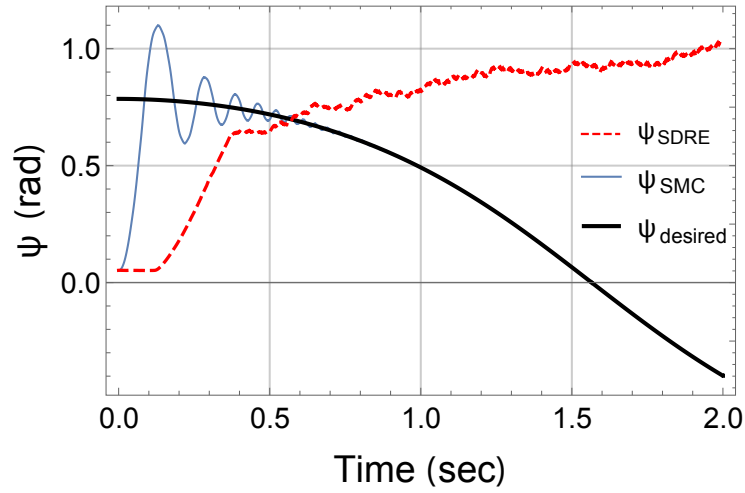
Figure 3.16: Orientation angles and 3D motion for jumping in a sinusoidal path without uncertainties in system.



(a)



(b)



(c)

Figure 3.17: Orientation angles and 3D motion for jumping in a sinusoidal path with uncertainties in system.

APPENDIX A

Equations of motion of Jumper

$$\ddot{\phi} = [I_{\phi}^{1}\,\dot{\phi}^{2} + I_{\phi}^{2}\,\dot{\theta}^{2} - I_{\phi}^{3}\,\dot{\psi}^{\,2} + I_{\phi}^{4}\,\dot{\phi}\,\dot{\theta} + I_{\phi}^{5}\,\dot{\phi}\,\dot{\psi} + I_{\phi}^{6}\,\dot{\theta}\,\dot{\psi} +$$
$$(I_{\phi}^{7} + I_{\phi}^{8})\nu M_T\, g\, z + \kappa_{\phi}^{1}\,\tau_{\phi} + \kappa_{\theta}^{1}\,\tau_{\theta} - \kappa_{\psi}^{1}\,\tau_{\psi}]/I_{den}$$
$$\ddot{\theta} = [I_{\theta}^{1}\,\dot{\phi}^{2} + I_{\theta}^{2}\,\dot{\theta}^{2} + I_{\theta}^{3}\,\dot{\psi}^{2} + I_{\theta}^{4}\,\dot{\phi}\,\dot{\theta} + I_{\theta}^{5}\,\dot{\phi}\,\dot{\psi} + I_{\theta}^{6}\,\dot{\theta}\,\dot{\psi} +$$
$$(I_{\theta}^{7} + I_{\theta}^{8})\nu M_T\, g\, z + \kappa_{\phi}^{2}\,\tau_{\phi} + \kappa_{\theta}^{2}\,\tau_{\theta} - \kappa_{\psi}^{2}\,\tau_{\psi}]/I_{den}$$
$$\ddot{\psi} = [I_{\psi}^{1}\,\dot{\phi}^{2} + I_{\psi}^{2}\,\dot{\theta}^{2} + I_{\psi}^{3}\,\dot{\psi}^{2} + I_{\psi}^{4}\,\dot{\phi}\,\dot{\theta} + I_{\psi}^{5}\,\dot{\phi}\,\dot{\psi} + I_{\psi}^{6}\,\dot{\theta}\,\dot{\psi} +$$
$$(I_{\psi}^{7} + I_{\psi}^{8})\nu M_T\, g\, z - \kappa_{\phi}^{3}\,\tau_{\phi} - \kappa_{\theta}^{3}\,\tau_{\theta} + \kappa_{\psi}^{3}\,\tau_{\psi}]/I_{den}$$
$$I_{den} = I_{xx}\left(I_{yz}^{2} - I_{yy}I_{zz}\right) + I_{xy}^{2}I_{zz} + 2I_{xy}I_{xz}I_{yz} + I_{xz}^{2}I_{yy}$$
$$I_{\phi}^{1} = I_{yz}(I_{xz}^{2} - I_{xy}^{2}) + I_{xy}I_{xz}(I_{zz} - I_{yy})$$
$$I_{\phi}^{2} = I_{yz}(I_{xy}^{2} - I_{yy}I_{zz} + I_{yz}^{2}) + I_{xy}I_{xz}I_{yy}$$

$$I_\phi^3 = I_{yz}(I_{xz}^2 - I_{yy}I_{zz} + I_{yz}^2) + I_{xy}I_{xz}I_{zz}$$

$$I_\phi^4 = I_{xy}I_{yz}(I_{yy} + I_{zz} - I_{xx}) + I_{xz}I_{yy}(I_{xx} - I_{yy} + I_{zz}) + 2I_{xz}I_{yz}^2$$

$$I_\phi^5 = I_{xy}I_{zz}(I_{xx} + I_{yy} - I_{zz}) + I_{xz}I_{yz}(I_{xx} - I_{yy} - I_{zz})$$
$$- 2I_{xy}I_{yz}^2$$

$$I_\phi^6 = I_{yy}\left(I_{xz}^2 + I_{yz}^2 + I_{zz}^2\right) - I_{zz}\left(I_{xy}^2 + I_{yy}^2 + I_{yz}^2\right)$$

$$I_\phi^7 = \left(I_{yy}I_{zz} - I_{yz}^2\right)\tan\phi,\ I_\phi^8 = (I_{xy}I_{zz} + I_{xz}I_{yz})\tan\theta\sec\phi$$

$$I_\theta^1 = I_{xz}(I_{xx}I_{zz} - I_{xy}^2 - I_{xz}^2) - I_{xx}I_{xy}I_{yz}$$

$$I_\theta^2 = I_{xy}I_{yz}(I_{xx} - I_{zz}) + I_{xz}(I_{xy}^2 - I_{yz}^2)$$

$$I_\theta^3 = I_{xz}\left(I_{yz}^2 - I_{xx}I_{zz} + I_{xz}^2\right) + I_{xy}I_{yz}I_{zz}$$

$$I_\theta^4 = I_{xx}I_{yz}(I_{yy} + I_{zz} - I_{xx}) - I_{xz}I_{xy}(I_{xx} - I_{yy} + I_{zz}) - 2I_{yz}I_{xz}^2$$

$$I_\theta^5 = I_{zz}\left(I_{xx}^2 + I_{xy}^2 + I_{xz}^2\right) - I_{xx}\left(I_{xz}^2 + I_{yz}^2 + I_{zz}^2\right)$$

$$I_\theta^6 = I_{xz}I_{yz}(I_{xx} - I_{yy} + I_{zz}) - I_{xy}I_{zz}(I_{xx} + I_{yy} - I_{zz}) + 2I_{xy}I_{xz}^2$$

$$I_\theta^7 = (I_{xy}I_{zz} + I_{xz}I_{yz})\tan\phi,\ I_\theta^8 = -\left(I_{xz}^2 - I_{xx}I_{zz}\right)\tan\theta\sec\phi$$

$$I_\psi^1 = I_{xy}\left(I_{xz}^2 - I_{xx}I_{yy} + I_{xy}^2\right) + I_{xx}I_{xz}I_{yz}$$

$$I_\psi^2 = I_{xy}\left(I_{xx}I_{yy} - I_{yz}^2 - I_{xy}^2\right) - I_{xz}I_{yy}I_{yz}$$

$$I_\psi^3 = I_{xz}I_{yz}(I_{yy} - I_{xx}) + I_{xy}\left(I_{yz}^2 - I_{xz}^2\right)$$

$$I_\psi^4 = I_{xx}\left(I_{xy}^2 + I_{yy}^2 + I_{yz}^2\right) - I_{yy}\left(I_{xx}^2 + I_{xy}^2 + I_{xz}^2\right)$$

$$I_\psi^5 = I_{xx}I_{yz}(I_{xx} - I_{yy} - I_{zz}) + I_{xy}I_{xz}(I_{xx} + I_{yy} - I_{zz}) +$$
$$2I_{xy}^2I_{yz}$$

$$I_\psi^6 = I_{xz}I_{yy}(I_{xx} - I_{yy} + I_{zz}) - I_{xy}I_{yz}(I_{xx} + I_{yy} - I_{zz}) -$$
$$2I_{xy}^2I_{xz}$$

$$I_\psi^7 = (I_{xy}I_{yz} + I_{xz}I_{yy})\tan\phi + (I_{xx}I_{yz} + I_{xy}I_{xz})\tan\theta\sec\phi$$

$$\kappa_\phi^1 = I_{yz}^2 + I_{yy}I_{zz},\ \kappa_\phi^2 = I_{xy}I_{zz} + I_{xz}I_{yz}$$

$$\kappa_\phi^3 = I_{xy}I_{yz} + I_{yy}I_{xz},\ \kappa_\theta^1 = I_{xz}I_{yz} - I_{xy}I_{zz}$$

$$\kappa_\theta^2 = I_{xz}^2 - I_{xx}I_{zz},\ \kappa_\theta^3 = I_{xx}I_{yz} + I_{xy}I_{xz}$$

$$\kappa_\psi^1 = I_{xy}I_{yz} + I_{xz}I_{yy},\ \kappa_\psi^2 = I_{xx}I_{yz} + I_{xy}I_{xz}$$

$$\kappa_\psi^3 = I_{xy}^2 - I_{xx}I_{yy}$$

SDC form of the equations of motion of Jumper

$$A(x) = \frac{1}{I_{den}} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & a_{26} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 & a_{46} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ a_{61} & a_{62} & a_{63} & a_{64} & 0 & a_{66} \end{bmatrix}$$

$$B(x) = \frac{1}{I_{den}} \begin{bmatrix} 0 & 0 & 0 \\ \kappa_\phi^1 & \kappa_\theta^1 & -\kappa_\psi^1 \\ 0 & 0 & 0 \\ \kappa_\phi^2 & \kappa_\theta^2 & -\kappa_\psi^2 \\ 0 & 0 & 0 \\ -\kappa_\phi^3 & -\kappa_\theta^3 & \kappa_\psi^3 \end{bmatrix}$$

$$Q(x) = \begin{bmatrix} 1+\theta^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1+\dot{\theta}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1+\phi^2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1+\dot{\phi}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+\psi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+\dot{\psi}^2 \end{bmatrix}$$

$$R(x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$a_{21} = \frac{I_{\phi}^{7} \nu M_T \, g \, z}{\phi},\ a_{22} = I_{\phi}^{1} \, \dot{\phi} + \frac{I_{\phi}^{4}}{2} \, \dot{\theta} + \frac{I_{\phi}^{5}}{2} \, \dot{\psi}$$

$$a_{23} = \frac{I_{\phi}^{8} \nu M_T \, g \, z}{\theta},\ a_{24} = I_{\phi}^{2} \, \dot{\theta} + \frac{I_{\phi}^{4}}{2} \, \dot{\phi} + \frac{I_{\phi}^{6}}{2} \, \dot{\psi}$$

$$a_{26} = -I_{\phi}^{3} \, \dot{\psi} + \frac{I_{\phi}^{5}}{2} \, \dot{\phi} + \frac{I_{\phi}^{6}}{2} \, \dot{\theta},\ a_{41} = \frac{I_{\theta}^{7} \nu M_T \, g \, z}{\theta}$$

$$a_{42} = I_{\theta}^{1} \, \dot{\phi} + \frac{I_{\theta}^{4}}{2} \, \dot{\theta} + \frac{I_{\theta}^{5}}{2} \, \dot{\psi},\ a_{43} = \frac{I_{\theta}^{8} \nu M_T \, g \, z}{\theta}$$

$$a_{44} = I_{\theta}^{2} \, \dot{\theta} + \frac{I_{\theta}^{4}}{2} \, \dot{\phi} + \frac{I_{\theta}^{6}}{2} \, \dot{\psi},\ a_{61} = \frac{I_{\psi}^{7} \nu M_T \, g \, z}{\theta}$$

$$a_{62} = I_{\psi}^{1} \, \dot{\phi} + \frac{I_{\psi}^{4}}{2} \, \dot{\theta} + \frac{I_{\psi}^{5}}{2} \, \dot{\psi},\ a_{63} = \frac{I_{\psi}^{8} \nu M_T \, g \, z}{\theta}$$

$$a_{64} = I_{\psi}^{2} \, \dot{\theta} + \frac{I_{\psi}^{4}}{2} \, \dot{\phi} + \frac{I_{\psi}^{6}}{2} \, \dot{\psi},\ a_{66} = I_{\psi}^{3} \, \dot{\psi} + \frac{I_{\psi}^{5}}{2} \, \dot{\phi} + \frac{I_{\psi}^{6}}{2} \, \dot{\theta}$$

BIBLIOGRAPHY

[1] M. Sitti, H. Ceylan, W. Hu, J. Giltinan, M. Turan, S. Yim, and E. Diller, "Biomedical applications of untethered mobile milli/microrobots," *Proceedings of the IEEE*, vol. 103, no. 2, pp. 205–224, 2015. 1

[2] E. Al Khatib, A. Bhattacharjee, P. Razzaghi, L. W. Rogowski, M. J. Kim, and Y. Hurmuzlu, "Magnetically actuated simple millirobots for complex navigation and modular assembly," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2958–2965, 2020. 1, 11, 15, 35, 39

[3] E. Al Khatib, "Control and locomotion of inertially and magnetically actuated multi-scale robotic systems," *PhD dissertation, Southern Methodist University*, 2020. 1

[4] U. K. Cheang, M. Dejan, J. Choi, and M. Kim, "Towards model-based control of achiral microswimmers," *ASME 2014 Dynamic Systems and Control Conference*, pp. V002T33A002–V002T33A002, 2014. 4

[5] D. Milutinovi and P. Lima, "Modeling and optimal centralized control of a large-size robotic population," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1280–1285, 2006. 9

[6] S. Shahrokhi, L. Lin, C. Ertel, M. Wan, and A. T. Becker, "Steering a swarm of particles using global inputs and swarm statistics," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 207–219, 2017. 9, 10

[7] X. Zhang, H. Kim, and M. J. Kim, "Design, implementation, and analysis of a 3-d magnetic tweezer system with high magnetic field gradient," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 3, pp. 680–687, 2018. 9

[8] M. K. Manshadi, M. Saadat, M. Mohammadi, M. Shamsi, M. Dejam, R. Kamali, and A. Sanati-Nezhad, "Delivery of magnetic micro/nanoparticles and magnetic-based drug/cargo into arterial flow for targeted therapy," *Drug delivery*, vol. 25, no. 1, pp. 1963–1973, 2018. 9

[9] M. K. Manshadi, M. Saadat, M. Mohammadi, R. Kamali, M. Shamsi, M. Naseh, and A. Sanati-Nezhad, "Magnetic aerosol drug targeting in lung cancer therapy using permanent magnet," *Drug delivery*, vol. 26, no. 1, pp. 120–128, 2019. 9

[10] H. Kim, J. Ali, U. K. Cheang, J. Jeong, J. S. Kim, and M. J. Kim, "Micro manipulation using magnetic microrobots," *Journal of Bionic Engineering*, vol. 13, no. 4, pp. 515–524, 2016. 10

[11] K. B. Yesin, K. Vollmers, and B. J. Nelson, "Modeling and control of untethered biomicrorobots in a fluidic environment using electromagnetic fields," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 527–536, 2006. 10

[12] S. Miyashita, S. Guitron, M. Ludersdorfer, C. R. Sung, and D. Rus, "An untethered miniature origami robot that self-folds, walks, swims, and degrades," *Robotics and Automation, 2015 IEEE International Conference on*, pp. 1490–1496, 2015. 10

[13] S. Miyashita, S. Guitron, K. Yoshida, S. Li, D. D. Damian, and D. Rus, "Ingestible, controllable, and degradable origami robot for patching stomach wounds," *Proceedings-IEEE International Conference on Robotics and Automation*, vol. 2016, pp. 909–916, 2016. 10

[14] A. W. Mahoney and J. J. Abbott, "Generating rotating magnetic fields with a single permanent magnet for propulsion of untethered magnetic devices in a lumen," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 411–420, 2014. 10

[15] ——, "5-dof manipulation of an untethered magnetic device in fluid using a single permanent magnet." *Robotics: Science and Systems*, 2014. 10

[16] D. Son, S. Yim, and M. Sitti, "A 5-d localization method for a magnetically manipulated untethered robot using a 2-d array of hall-effect sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 708–716, 2016. 10

[17] J. Kuthan, M. Juřik, and F. Mach, "Magnetic actuation of multiple robots by the coplanar coils system," *2019 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, pp. 1–5, 2019. 10

[18] A. Mahadev, D. Krupke, S. P. Fekete, and A. T. Becker, "Mapping and coverage with a particle swarm controlled by uniform inputs," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1097–1104, 2017. 10

[19] P. Joshi, J. Leclerc, D. Bao, and A. T. Becker, "Motion-planning using rrts for a swarm of robots controlled by global inputs," *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 1163–1168, 2019. 10

[20] S. Shahrokhi, A. Mahadev, and A. T. Becker, "Algorithms for shaping a particle swarm with a shared input by exploiting non-slip wall contacts," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4304–4311, 2017. 10

[21] S. Shahrokhi, J. Shi, B. Isichei, and A. T. Becker, "Exploiting nonslip wall contacts to position two particles using the same control input," *IEEE Transactions on Robotics*, 2019. 10, 39

[22] B. R. Donald, C. G. Levey, and I. Paprotny, "Planar microassembly by parallel actuation of mems microrobots," *Journal of Microelectromechanical Systems*, vol. 17, no. 4, pp. 789–808, 2008. 10

[23] S. Floyd, E. Diller, C. Pawashe, and M. Sitti, "Control methodologies for a heterogeneous group of untethered magnetic micro-robots," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1553–1565, 2011. 10

[24] E. Diller, J. Giltinan, and M. Sitti, "Independent control of multiple magnetic microrobots in three dimensions," *The International Journal of Robotics Research*, vol. 32, no. 5, pp. 614–631, 2013. 10, 39

[25] T. H. Vose, P. Umbanhowar, and K. M. Lynch, "Sliding manipulation of rigid bodies on a controlled 6-dof plate," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 819–838, 2012. 10

[26] S. Shahrokhi, L. Lin, and A. T. Becker, "Planar orientation control and torque maximization using a swarm with global inputs," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1980–1987, 2019. 10

[27] X. Dong and M. Sitti, "Controlling two-dimensional collective formation and cooperative behavior of magnetic microrobot swarms," *The International Journal of Robotics Research*, p. 0278364920903107, 2020. 10

[28] J. Zoghzoghy, J. Zhao, and Y. Hurmuzlu, "Modeling, design, and implementation of a baton robot with double-action inertial actuation," *Mechatronics*, vol. 29, pp. 1–12, 2015. 11, 131

[29] M. Kashki, S. Ercan, and Y. Hurmuzlu, "Pivot walking of an inertially actuated robot," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1152–1162, 2016. 11, 131, 136, 139

[30] P. Razzaghi, E. Al Khatib, and Y. Hurmuzlu, "Nonlinear dynamics and control of an inertially actuated jumper robot," *Nonlinear Dynamics*, vol. 97, no. 1, pp. 161–176, 2019. 11, 71, 119

[31] J. J. Abbott, "Parametric design of tri-axial nested helmholtz coils," *Review of scientific instruments*, vol. 86, no. 5, p. 054701, 2015. 13

[32] S. Palagi and P. Fischer, "Bioinspired microrobots," *Nature Reviews Materials*, vol. 3, no. 6, p. 113, 2018. 39

[33] A. Aghakhani, O. Yasa, P. Wrede, and M. Sitti, "Acoustically powered surface-slipping mobile microrobots," *Proceedings of the National Academy of Sciences*, vol. 117, no. 7, pp. 3469–3477, 2020. 39

[34] S. Shahrokhi and A. T. Becker, "Object manipulation and position control using a swarm with global inputs," *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 561–566, 2016. 39

[35] V. Sridhar, B.-W. Park, and M. Sitti, "Light-driven janus hollow mesoporous tio2–au microswimmers," *Advanced Functional Materials*, vol. 28, no. 25, p. 1704902, 2018. 39

[36] H. Shahsavan, A. Aghakhani, H. Zeng, Y. Guo, Z. S. Davidson, A. Priimagi, and M. Sitti, "Bioinspired underwater locomotion of light-driven liquid crystal gels," *Proceedings of the National Academy of Sciences*, vol. 117, no. 10, pp. 5125–5133, 2020. 39

[37] J. J. Abbott, K. E. Peyer, M. C. Lagomarsino, L. Zhang, L. Dong, I. K. Kaliakatsos, and B. J. Nelson, "How should microrobots swim?" *The international journal of Robotics Research*, vol. 28, no. 11-12, pp. 1434–1447, 2009. 39

[38] H.-W. Tung, K. E. Peyer, D. F. Sargent, and B. J. Nelson, "Noncontact manipulation using a transversely magnetized rolling robot," *Applied Physics Letters*, vol. 103, no. 11, p. 114101, 2013. 39

[39] U. Bozuyuk, O. Yasa, I. C. Yasa, H. Ceylan, S. Kizilel, and M. Sitti, "Light-triggered drug release from 3d-printed magnetic chitosan microswimmers," *ACS nano*, vol. 12, no. 9, pp. 9617–9625, 2018. 39

[40] E. Diller, J. Zhuang, G. Zhan Lum, M. R. Edwards, and M. Sitti, "Continuously distributed magnetization profile for millimeter-scale elastomeric undulatory swimming," *Applied Physics Letters*, vol. 104, no. 17, p. 174101, 2014. 39

[41] W. Hu, G. Z. Lum, M. Mastrangeli, and M. Sitti, "Small-scale soft-bodied robot with multimodal locomotion," *Nature*, vol. 554, no. 7690, pp. 81–85, 2018. 39

[42] C. Pawashe, S. Floyd, and M. Sitti, "Modeling and experimental characterization of an untethered magnetic micro-robot," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 1077–1094, 2009. 39

[43] A. de Oliveira Barros and J. Yang, "A review of magnetically actuated milli/micro-scale robots locomotion and features," *Critical Reviews^{TM} in Biomedical Engineering*, vol. 47, no. 5, 2019. 39

[44] C. Bi, E. E. Niedert, G. Adam, E. Lambert, L. Solorio, C. J. Goergen, and D. J. Cappelleri, "Tumbling magnetic microrobots for biomedical applications," *2019 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, pp. 1–6, 2019. 39

[45] J. Zhang and E. Diller, "Untethered miniature soft robots: Modeling and design of a millimeter-scale swimming magnetic sheet," *Soft robotics*, vol. 5, no. 6, pp. 761–776, 2018. 39

[46] X. Dong and M. Sitti, "Planning spin-walking locomotion for automatic grasping of microobjects by an untethered magnetic microgripper," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6612–6618, 2017. 40

[47] Y. Hurmuzlu, F. GéNot, and B. Brogliato, "Modeling, stability and control of biped robots—a general framework," *Automatica*, vol. 40, no. 10, pp. 1647–1664, 2004. 45, 71, 119, 145

[48] G. W. Gabriel, T. R. Gonçalves, and J. C. Geromel, "Optimal and robust sampled-data control of markov jump linear systems: A differential lmi approach," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 3054–3060, 2018. 70

[49] B. P. Van Parys, D. Kuhn, P. J. Goulart, and M. Morari, "Distributionally robust control of constrained stochastic systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 430–442, 2016. 70

[50] X. Zhang, M. Kamgarpour, A. Georghiou, P. Goulart, and J. Lygeros, "Robust optimal control with adjustable uncertainty sets," *Automatica*, vol. 75, pp. 249–259, 2017. 70

[51] M. H. Terra, J. P. Cerri, and J. Y. Ishihara, "Optimal robust linear quadratic regulator for systems subject to uncertainties," *IEEE Transactions on Automatic Control*, vol. 59, no. 9, pp. 2586–2591, 2014. 70

[52] C. Hu, H. Jing, R. Wang, F. Yan, and M. Chadli, "Robust h-inf output-feedback control for path following of autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 70, pp. 414–427, 2016. 70

[53] X. Yang and H. He, "Self-learning robust optimal control for continuous-time nonlinear systems with mismatched disturbances," *Neural Networks*, vol. 99, pp. 19–30, 2018. 70

[54] A. C. Satici, H. Poonawala, and M. W. Spong, "Robust optimal control of quadrotor uavs," *IEEE Access*, vol. 1, pp. 79–93, 2013. 70

[55] Q. Zhang, D. Zhao, and D. Wang, "Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 1, pp. 37–50, 2018. 70

[56] X. Wang, E. E. Yaz, S. C. Schneider, and Y. I. Yaz, "H2-h infinity control of continuous-time nonlinear systems using the state-dependent riccati equation approach," *Systems Science & Control Engineering*, vol. 5, no. 1, pp. 224–231, 2017. 70, 78, 109

[57] T. Bian and Z.-P. Jiang, "Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design," *Automatica*, vol. 71, pp. 348–360, 2016. 70

[58] H. Ma, Z. Wang, D. Wang, D. Liu, P. Yan, and Q. Wei, "Neural-network-based distributed adaptive robust control for a class of nonlinear multiagent systems with time delays and external noises," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 750–758, 2015. 70

[59] D. Telen, M. Vallerio, L. Cabianca, B. Houska, J. Van Impe, and F. Logist, "Approximate robust optimization of nonlinear systems under parametric uncertainty and process noise," *Journal of Process Control*, vol. 33, pp. 140–154, 2015. 70

[60] P. Razzaghi, E. A. Khatib, and Y. Hurmuzlu, "Robust quadratic gaussian control of continuous-time nonlinear systems," *arXiv preprint arXiv:1912.06717*, 2019. 71, 102

[61] J. R. Cloutier, C. N. D'Souza, and C. P. Mracek, "Nonlinear regulation and nonlinear h_∞ control via the state-dependent riccati equation technique: Part 1, theory," *Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace*, pp. 117–130, 1996. 71

[62] Y. Hurmuzlu, "Dynamics and control of bipedal robots," *Control Problems in Robotics and Automation*, pp. 105–117, 1998. 71

[63] E. Borzova and Y. Hurmuzlu, "Passively walking five-link robot," *Automatica*, vol. 40, no. 4, pp. 621–629, 2004. 71

[64] T. Çimen, "State-dependent riccati equation (sdre) control: A survey," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3761–3775, 2008. 71

[65] B. Friedland, *Advanced control system design*. Prentice-Hall, Inc., 1995. 72

[66] A. Sinha, *Linear systems: optimal and robust control*. CRC press, 2007. 73

[67] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4. 75

[68] T. Iwasaki and R. Skelton, "Parametrization of all stabilizing controllers via quadratic lyapunov functions," *Journal of Optimization Theory and Applications*, vol. 85, no. 2, pp. 291–307, 1995. 77

[69] T. Çimen, "Systematic and effective design of nonlinear feedback controllers via the state-dependent riccati equation (sdre) method," *Annual Reviews in control*, vol. 34, no. 1, pp. 32–51, 2010. 79

[70] J. R. Cloutier and D. T. Stansbery, "The capabilities and art of state-dependent riccati equation-based design," *American Control Conference, 2002. Proceedings of the 2002*, vol. 1, pp. 86–91, 2002. 79, 143

[71] L.-G. Lin and M. Xin, "Computational enhancement of the sdre scheme: General theory and robotic control system," *IEEE Transactions on Robotics*, 2020. 79

[72] Y. Feng, B. D. Anderson, and M. Rotkowitz, "A game theoretic algorithm to compute local stabilizing solutions to hjbi equations in nonlinear h-inf control," *Automatica*, vol. 45, no. 4, pp. 881–888, 2009. 79

[73] M. Aliyu, "An iterative relaxation approach to the solution of the hamilton-jacobi-bellman-isaacs equation in nonlinear optimal control," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 360–366, 2017. 80

[74] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems using an online hamilton-jacobi-isaacs formulation," *49th IEEE Conference on Decision and Control (CDC)*, pp. 3048–3053, 2010. 80

[75] A. Heydari and S. N. Balakrishnan, "Fixed-final-time optimal control of nonlinear systems with terminal constraints," *Neural Networks*, vol. 48, pp. 61–71, 2013. 82

[76] ——, "Optimal switching between autonomous subsystems," *Journal of the Franklin Institute*, vol. 351, no. 5, pp. 2675–2690, 2014. 82

[77] ——, "Global optimality of approximate dynamic programming and its use in non-convex function minimization," *Applied Soft Computing*, vol. 24, pp. 291–303, 2014. 83

[78] J. Missel and D. Mortari, "Removing space debris through sequential captures and ejections," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, pp. 743–752, 2013. 94

[79] C. R. Phipps, K. L. Baker, S. B. Libby, D. A. Liedahl, S. S. Olivier, L. D. Pleasance, A. Rubenchik, J. E. Trebes, E. V. George, B. Marcovici *et al.*, "Removing orbital debris with lasers," *Advances in Space Research*, vol. 49, no. 9, pp. 1283–1300, 2012. 94

[80] L. DeLuca, F. Bernelli, F. Maggi, P. Tadini, C. Pardini, L. Anselmo, M. Grassi, D. Pavarin, A. Francesconi, F. Branz *et al.*, "Active space debris removal by a hybrid propulsion module," *Acta Astronautica*, vol. 91, pp. 20–33, 2013. 94

[81] C. Bombardelli and J. Pelaez, "Ion beam shepherd for contactless space debris removal," *Journal of guidance, control, and dynamics*, vol. 34, no. 3, pp. 916–920, 2011. 94

[82] D. G. Stuart, "Guidance and control for cooperative tether-mediated orbital rendezvous," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 1102–1108, 1990. 94

[83] M. L. Cosmo, E. C. Lorenzini *et al.*, *Tethers in space handbook*. National Aeronautics and Space Administration Washington, USA, 1997. 94

[84] V. Aslanov and V. Yudintsev, "Dynamics of large space debris removal using tethered space tug," *Acta Astronautica*, vol. 91, pp. 149–156, 2013. 94, 95

[85] P. Huang, D. Wang, F. Zhang, Z. Meng, and Z. Liu, "Postcapture robust nonlinear control for tethered space robot with constraints on actuator and velocity of space tether," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 16, pp. 2824–2841, 2017. 95

[86] M. A. A. Hallaj and N. Assadian, "Sliding mode control of electromagnetic tethered satellite formation," *Advances in Space Research*, vol. 58, no. 4, pp. 619–634, 2016. 95

[87] L. Jasper and H. Schaub, "Input shaped large thrust maneuver with a tethered debris object," *Acta Astronautica*, vol. 96, pp. 128–137, 2014. 95

[88] Z. Meng, P. Huang, and J. Guo, "Approach modeling and control of an autonomous maneuverable space net," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 6, pp. 2651–2661, 2017. 95

[89] R. Benvenuto, S. Salvi, and M. Lavagna, "Dynamics analysis and gnc design of flexible systems for space debris active removal," *Acta Astronautica*, vol. 110, pp. 247–265, 2015. 95

[90] S.-I. Nishida, S. Kawamoto, Y. Okawa, F. Terui, and S. Kitamura, "Space debris removal system using a small satellite," *Acta Astronautica*, vol. 65, no. 1-2, pp. 95–102, 2009. 95

[91] V. S. Aslanov and V. V. Yudintsev, "Behavior of tethered debris with flexible appendages," *Acta Astronautica*, vol. 104, no. 1, pp. 91–98, 2014. 95

[92] ——, "Dynamics, analytical solutions and choice of parameters for towed space debris with flexible appendages," *Advances in Space Research*, vol. 55, no. 2, pp. 660–667, 2015. 95

[93] R. Hoyt and R. Forward, "The terminator tether-autonomous deorbit of leo spacecraft for space debris mitigation," *38th Aerospace Sciences Meeting and Exhibit*, p. 329, 2000. 95

[94] M. Shan, J. Guo, and E. Gill, "Deployment dynamics of tethered-net for space debris removal," *Acta Astronautica*, vol. 132, pp. 293–302, 2017. 95

[95] Y. Zhao, P. Huang, and F. Zhang, "Capture dynamics and net closing control for tethered space net robot," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 1, pp. 199–208, 2018. 95

[96] F. Zhang and P. Huang, "Releasing dynamics and stability control of maneuverable tethered space net," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 983–993, 2016. 95

[97] F. Zhang, P. Huang, Z. Meng, Y. Zhang, and Z. Liu, "Dynamics analysis and controller design for maneuverable tethered space net robot," *Journal of guidance, control, and dynamics*, vol. 40, no. 11, pp. 2828–2843, 2017. 95

[98] R. Qi, A. K. Misra, and Z. Zuo, "Active debris removal using double-tethered space-tug system," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 3, pp. 722–730, 2016. 95

[99] P. Razzaghi, E. Al Khatib, S. Bakhtiari, and Y. Hurmuzlu, "Real time control of tethered satellite systems to de-orbit space debris," *Aerospace Science and Technology*, p. 106379, 2020. 95

[100] P. Razzaghi and N. Assadian, "Study of the triple-mass tethered satellite system under aerodynamic drag and j2 perturbations," *Advances in Space Research*, vol. 56, no. 10, pp. 2141–2150, 2015. 97, 116, 118

[101] K. Moe, M. M. Moe, and S. D. Wallace, "Drag coefficients of spheres in free-molecular flow," *Advances in the Astronautical Sciences*, vol. 93, pp. 391–406, 1996. 98, 99

[102] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001, vol. 12. 99, 117

[103] J. Moreno-Valenzuela, C. Aguilar-Avelar, S. Puga-Guzmán, and V. Santibáñez, "Two adaptive control strategies for trajectory tracking of the inertia wheel pendulum: neural networks vis à vis model regressor," *Intelligent Automation & Soft Computing*, vol. 23, no. 1, pp. 63–73, 2017. 109

[104] S. R. Nekoo, "Model reference adaptive state-dependent riccati equation control of nonlinear uncertain systems: Regulation and tracking of free-floating space manipulators," *Aerospace Science and Technology*, vol. 84, pp. 348–360, 2019. 109

[105] Z. Ma and G. Sun, "Adaptive hierarchical sliding mode control with input saturation for attitude regulation of multi-satellite tethered system," *The Journal of the Astronautical Sciences*, vol. 64, no. 2, pp. 207–230, 2017. 109

[106] B. Yu, D. Jin, and H. Wen, "Nonlinear dynamics of flexible tethered satellite system subject to space environment," *Applied Mathematics and Mechanics*, vol. 37, no. 4, pp. 485–500, 2016. 110, 118

[107] G. B. Palmerini and M. Sabatini, "Dynamics and control of low-altitude formations," *Acta Astronautica*, vol. 61, no. 1-6, pp. 298–311, 2007. 110

[108] P. Razzaghi, E. Al Khatib, K. S. Alluhydan, and Y. Hurmuzlu, "$h_2 - h_\infty$ model reference adaptive control of tethered satellite system," *2020 IEEE Aerospace Conference*, pp. 1–8, 2020. 110

[109] P. Williams, S. Yeo, and C. Blanksby, "Heating and modeling effects in tethered aerocapture missions," *Journal of guidance, control, and dynamics*, vol. 26, no. 4, pp. 643–654, 2003. 118

[110] G. Avanzini and M. Fedi, "Effects of eccentricity of the reference orbit on multi-tethered satellite formations," *Acta Astronautica*, vol. 94, no. 1, pp. 338–350, 2014. 118

[111] H. Nijmeijer and A. Van der Schaft, *Nonlinear dynamical control systems*. Springer, 1990, vol. 175. 118, 136

[112] A. Sayyad, B. Seth, and P. Seshu, "Single-legged hopping robotics research—a review," *Robotica*, vol. 25, no. 5, pp. 587–613, 2007. 130

[113] J. Burdick and P. Fiorini, "Minimalist jumping robots for celestial exploration," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 653–674, 2003. 130

[114] B. G. Lambrecht, A. D. Horchler, and R. D. Quinn, "A small, insect-inspired robot that runs and jumps," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 1240–1245, 2005. 131

[115] S. Dubowsky, S. Kesner, J.-S. Plante, and P. Boston, "Hopping mobility concept for search and rescue robots," *Industrial Robot: An International Journal*, vol. 35, no. 3, pp. 238–245, 2008. 131

[116] C.-Y. Chan and Y.-C. Liu, "Towards a walking, turning, and jumping quadruped robot with compliant mechanisms," *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*, pp. 614–620, 2016. 131

[117] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 932–950, 2013. 131

[118] H. Tsukagoshi, M. Sasaki, A. Kitagawa, and T. Tanaka, "Design of a higher jumping rescue robot with the optimized pneumatic drive," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 1276–1283, 2005. 131

[119] N. W. Bartlett, M. T. Tolley, J. T. Overvelde, J. C. Weaver, B. Mosadegh, K. Bertoldi, G. M. Whitesides, and R. J. Wood, "A 3d-printed, functionally graded soft robot powered by combustion," *Science*, vol. 349, no. 6244, pp. 161–165, 2015. 131

[120] B. Brogliato and A. Z. Rio, "On the control of complementary-slackness juggling mechanical systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 2, pp. 235–246, 2000. 131

[121] B. Brogliato, M. Mabrouk, and A. Z. Rio, "On the controllability of linear juggling mechanical systems," *Systems & control letters*, vol. 55, no. 4, pp. 350–367, 2006. 131

[122] Y. Hurmuzlu, "Dynamics of bipedal gait: Part i—objective functions and the contact event of a planar five-link biped," *Journal of Applied Mechanics*, vol. 60, no. 2, pp. 331–336, 1993. 131

[123] ——, "Dynamics of bipedal galt: Part ii-stability analysis of a planar five-link biped," *Transactions-American society of mechanical engineers of applied mechanics*, vol. 60, pp. 337–337, 1993. 131

[124] M. Gharib, A. Tavakoli, and Y. Hurmuzlu, "Kinematics and dynamics of a sliding/bouncing two mass system," *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 237–246, 2011. 131

[125] A. Tavakoli, M. Gharib, and Y. Hurmuzlu, "Collision of two mass baton with massive external surfaces," *Journal of applied mechanics*, vol. 79, no. 5, p. 051019, 2012. 131

[126] A. Tavakoli and Y. Hurmuzlu, "Robotic locomotion of three generations of a family tree of dynamical systems. part i: Passive gait patterns," *Nonlinear Dynamics*, vol. 73, no. 3, pp. 1969–1989, 2013. 131

[127] J. Zoghzoghy and Y. Hurmuzlu, "Dynamics, stability, and experimental results for a baton robot with double-action inertial actuation," *International Journal of Dynamics and Control*, pp. 1–19, 2017. 131

[128] A. Tavakoli and Y. Hurmuzlu, "Robotic locomotion of three generations of a family tree of dynamical systems. part ii: Impulsive control of gait patterns," *Nonlinear Dynamics*, vol. 73, no. 3, pp. 1991–2012, 2013. 131

[129] M. Kashki, J. Zoghzoghy, and Y. Hurmuzlu, "Adaptive control of inertially actuated bouncing robot," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 5, pp. 2196–2207, 2017. 131, 132, 146

[130] E. Al Khatib, W. M. Al-Masri, S. Mukhopadhyay, M. A. Jaradat, and M. Abdel-Hafez, "A comparison of adaptive trajectory tracking controllers for wheeled mobile robots," *Mechatronics and its Applications (ISMA), 2015 10th International Symposium on*, pp. 1–6, 2015. 131

[131] S. Nateghi and Y. Shtessel, "Robust stabilization of linear differential inclusion using adaptive sliding mode control," *2018 Annual American Control Conference (ACC)*, pp. 5327–5331, 2018. 131

[132] E. I. Al Khatib, M. A. Jaradat, M. Abdel-Hafez, and M. Roigari, "Multiple sensor fusion for mobile robot localization and navigation using the extended kalman filter," *Mechatronics and its Applications (ISMA), 2015 10th International Symposium on*, pp. 1–5, 2015. 131

[133] T.-H. Chang and Y. Hurmuzlu, "Sliding control without reaching phase and its application to bipedal locomotion," *Transactions-American society of mechanical engineers journal of dynamic systems measurments and control*, vol. 115, pp. 447–447, 1993. 131

[134] J.-M. Yang and J.-H. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots," *IEEE Transactions on robotics and automation*, vol. 15, no. 3, pp. 578–587, 1999. 131

[135] R. Cui, L. Chen, C. Yang, and M. Chen, "Extended state observer-based integral sliding mode control for an underwater robot with unknown disturbances and uncertain nonlinearities," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6785–6795, 2017. 131

[136] S. Kim and S. Kwon, "Nonlinear optimal control design for underactuated two-wheeled inverted pendulum mobile platform," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2803–2808, 2017. 132

[137] M. Korayem, A. Zehfroosh, H. Tourajizadeh, and S. Manteghi, "Optimal motion planning of non-linear dynamic systems in the presence of obstacles and moving boundaries using sdre: application on cable-suspended robot," *Nonlinear Dynamics*, vol. 76, no. 2, pp. 1423–1441, 2014. 132

[138] O. Okubo, E. Nakano, and M. Handa, "Design of a jumping machine using self-energizing spring," *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 1, pp. 186–191, 1996. 132

[139] R. Hayashi and S. Tsujio, "High-performance jumping movements by pendulum-type jumping machines," *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 2, pp. 722–727, 2001. 132

[140] K. Alluhydan, P. Razzaghi, and Y. Hurmuzlu, "On planar impacts of cylinders and balls," *Journal of Applied Mechanics*, vol. 86, no. 7, 2019. 135

[141] P. Razzaghi, K. Alluhydan, and Y. Hurmuzlu, "Planar impacts in hybrid chains of cylinders and balls," *Journal of Applied Mechanics*, vol. 86, no. 12, 2019. 135

[142] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1. 141

[143] Y. Hürmüzlü and G. D. Moskowitz, "Bipedal locomotion stabilized by impact and switching: I. two-and three-dimensional, three-element models," *Dynamics and Stability of Systems*, vol. 2, no. 2, pp. 73–96, 1987. 145

[144] ——, "Bipedal locomotion stabilized by impact and switching: Ii. structural stability analysis of a four-element bipedal locomotion model," *Dynamics and Stability of Systems*, vol. 2, no. 2, pp. 97–112, 1987. 145